Bao-Liang Lu
Liqing Zhang
James Kwok (Eds.)

LNCS 7063

Neural Information Processing

18th International Conference, ICONIP 2011
Shanghai, China, November 2011
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Computer Science 7063

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Alfred Kobsa
University of California, Irvine, CA, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
TU Dortmund University, Germany

Madhu Sudan
Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Gerhard Weikum
Max Planck Institute for Informatics, Saarbruecken, Germany

Bao-Liang Lu Liqing Zhang
James Kwok (Eds.)

Neural Information Processing

18th International Conference, ICONIP 2011
Shanghai, China, November 13-17, 2011
Proceedings, Part II

Springer

Volume Editors

Bao-Liang Lu
Shanghai Jiao Tong University
Department of Computer Science and Engineering
800, Dongchuan Road, Shanghai 200240, China
E-mail: bllu@sjtu.edu.cn

Liqing Zhang
Shanghai Jiao Tong University
Department of Computer Science and Engineering
800, Dongchuan Road, Shanghai 200240, China
E-mail: zhang-lq@cs.sjtu.edu.cn

James Kwok
The Hong Kong University of Science and Technology
Department of Computer Science and Engineering
Clear Water Bay, Kowloon, Hong Kong, China
E-mail: jamesk@cse.ust.hk

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-24957-0 e-ISBN 978-3-642-24958-7
DOI 10.1007/978-3-642-24958-7
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011939737

CR Subject Classification (1998): F.1, I.2, I.4-5, H.3-4, G.3, J.3, C.1.3, C.3

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

© Springer-Verlag Berlin Heidelberg 2011
This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.
The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This book and its sister volumes constitute the proceedings of the 18th International Conference on Neural Information Processing (ICONIP 2011) held in Shanghai, China, during November 13–17, 2011. ICONIP is the annual conference of the Asia Pacific Neural Network Assembly (APNNA). ICONIP aims to provide a high-level international forum for scientists, engineers, educators, and students to address new challenges, share solutions, and discuss future research directions in neural information processing and real-world applications.

The scientific program of ICONIP 2011 presented an outstanding spectrum of over 260 research papers from 42 countries and regions, emerging from multidisciplinary areas such as computational neuroscience, cognitive science, computer science, neural engineering, computer vision, machine learning, pattern recognition, natural language processing, and many more to focus on the challenges of developing future technologies for neural information processing. In addition to the contributed papers, we were particularly pleased to have 10 plenary speeches by world-renowned scholars: Shun-ichi Amari, Kunihiko Fukushima, Aike Guo, Lei Xu, Jun Wang, DeLiang Wang, Derong Liu, Xin Yao, Soo-Young Lee, and Nikola Kasabov. The program also includes six excellent tutorials by David Cai, Irwin King, Pei-Ji Liang, Hiroshi Mamitsuka, Ming Zhou, Hang Li, and Shanfeng Zhu. The conference was followed by three post-conference workshops held in Hangzhou, on November 18, 2011: "ICONIP2011Workshop on Brain – Computer Interface and Applications," organized by Bao-Liang Lu, Liqing Zhang, and Chin-Teng Lin; "The 4th International Workshop on Data Mining and Cybersecurity," organized by Paul S. Pang, Tao Ban, Youki Kadobayashi, and Jungsuk Song; and "ICONIP 2011 Workshop on Recent Advances in Nature-Inspired Computation and Its Applications," organized by Xin Yao and Shan He.

The ICONIP 2011 organizers would like to thank all special session organizers for their effort and time high enriched the topics and program of the conference. The program included the following 13 special sessions: "Advances in Computational Intelligence Methods-Based Pattern Recognition," organized by Kai-Zhu Huang and Jun Sun; "Biologically Inspired Vision and Recognition," organized by Jun Miao, Libo Ma, Liming Zhang, Juyang Weng and Xilin Chen; "Biomedical Data Analysis," organized by Jie Yang and Guo-Zheng Li; "Brain Signal Processing," organized by Jian-Ting Cao, Tomasz M. Rutkowski, Toshihisa Tanaka, and Liqing Zhang; "Brain-Realistic Models for Learning, Memory and Embodied Cognition," organized by Huajin Tang and Jun Tani; "Clifford Algebraic Neural Networks," organized by Tohru Nitta and Yasuaki Kuroe; "Combining Multiple Learners," organized by Younès Bennani, Nistor Grozavu, Mohamed Nadif, and Nicoleta Rogovschi; "Computational Advances in Bioinformatics," organized by Jonathan H. Chan; "Computational-Intelligent Human–Computer Interaction," organized by Chin-Teng Lin, Jyh-Yeong Chang,

John Kar-Kin Zao, Yong-Sheng Chen, and Li-Wei Ko; "Evolutionary Design and Optimization," organized by Ruhul Sarker and Mao-Lin Tang; "Human-Originated Data Analysis and Implementation," organized by Hyeyoung Park and Sang-Woo Ban; "Natural Language Processing and Intelligent Web Information Processing," organized by Xiao-Long Wang, Rui-Feng Xu, and Hai Zhao; and "Integrating Multiple Nature-Inspired Approaches," organized by Shan He and Xin Yao.

The ICONIP 2011 conference and post-conference workshops would not have achieved their success without the generous contributions of many organizations and volunteers. The organizers would also like to express sincere thanks to APNNA for the sponsorship, to the China Neural Networks Council, International Neural Network Society, and Japanese Neural Network Society for their technical co-sponsorship, to Shanghai Jiao Tong University for its financial and logistic supports, and to the National Natural Science Foundation of China, Shanghai Hyron Software Co., Ltd., Microsoft Research Asia, Hitachi (China) Research & Development Corporation, and Fujitsu Research and Development Center, Co., Ltd. for their financial support.

We are very pleased to acknowledge the support of the conference Advisory Committee, the APNNA Governing Board and Past Presidents for their guidance, and the members of the International Program Committee and additional reviewers for reviewing the papers. Particularly, the organizers would like to thank the proceedings publisher, Springer, for publishing the proceedings in the *Lecture Notes in Computer Science Series*. We want to give special thanks to the Web managers, Haoyu Cai and Dong Li, and the publication team comprising Li-Chen Shi, Yong Peng, Cong Hui, Bing Li, Dan Nie, Ren-Jie Liu, Tian-Xiang Wu, Xue-Zhe Ma, Shao-Hua Yang, Yuan-Jian Zhou and Cong Xie for checking the accepted papers in a short period of time. Last but not least, the organizers would like to thank all the authors, speakers, audience, and volunteers.

November 2011

Bao-Liang Lu
Liqing Zhang
James Kwok

ICONIP 2011 Organization

Organizer

Shanghai Jiao Tong University

Sponsor

Asia Pacific Neural Network Assembly

Financial Co-sponsors

Shanghai Jiao Tong University
National Natural Science Foundation of China
Shanghai Hyron Software Co., Ltd.
Microsoft Research Asia
Hitachi (China) Research & Development Corporation
Fujitsu Research and Development Center, Co., Ltd.

Technical Co-sponsors

China Neural Networks Council
International Neural Network Society
Japanese Neural Network Society

Honorary Chair

Shun-ichi Amari	Brain Science Institute, RIKEN, Japan

Advisory Committee Chairs

Shoujue Wang	Institute of Semiconductors, Chinese Academy of Sciences, China
Aike Guo	Institute of Neuroscience, Chinese Academy of Sciences, China
Liming Zhang	Fudan University, China

Advisory Committee Members

Sabri Arik	Istanbul University, Turkey
Jonathan H. Chan	King Mongkut's University of Technology, Thailand
Wlodzislaw Duch	Nicolaus Copernicus University, Poland
Tom Gedeon	Australian National University, Australia
Yuzo Hirai	University of Tsukuba, Japan
Ting-Wen Huang	Texas A&M University, Qatar
Akira Hirose	University of Tokyo, Japan
Nik Kasabov	Auckland University of Technology, New Zealand
Irwin King	The Chinese University of Hong Kong, Hong Kong
Weng-Kin Lai	MIMOS, Malaysia
Min-Ho Lee	Kyungpoor National University, Korea
Soo-Young Lee	Korea Advanced Institute of Science and Technology, Korea
Andrew Chi-Sing Leung	City University of Hong Kong, Hong Kong
Chin-Teng Lin	National Chiao Tung University, Taiwan
Derong Liu	University of Illinois at Chicago, USA
Noboru Ohnishi	Nagoya University, Japan
Nikhil R. Pal	Indian Statistical Institute, India
John Sum	National Chung Hsing University, Taiwan
DeLiang Wang	Ohio State University, USA
Jun Wang	The Chinese University of Hong Kong, Hong Kong
Kevin Wong	Murdoch University, Australia
Lipo Wang	Nanyang Technological University, Singapore
Xin Yao	University of Birmingham, UK
Liqing Zhang	Shanghai Jiao Tong University, China

General Chair

Bao-Liang Lu	Shanghai Jiao Tong University, China

Program Chairs

Liqing Zhang	Shanghai Jiao Tong University, China
James T.Y. Kwok	Hong Kong University of Science and Technology, Hong Kong

Organizing Chair

Hongtao Lu	Shanghai Jiao Tong University, China

Workshop Chairs

Guangbin Huang	Nanyang Technological University, Singapore
Jie Yang	Shanghai Jiao Tong University, China
Xiaorong Gao	Tsinghua University, China

Special Sessions Chairs

Changshui Zhang	Tsinghua University, China
Akira Hirose	University of Tokyo, Japan
Minho Lee	Kyungpoor National University, Korea

Tutorials Chair

Si Wu	Institute of Neuroscience, Chinese Academy of Sciences, China

Publications Chairs

Yuan Luo	Shanghai Jiao Tong University, China
Tianfang Yao	Shanghai Jiao Tong University, China
Yun Li	Nanjing University of Posts and Telecommunications, China

Publicity Chairs

Kazushi Ikeda	Nara Institute of Science and Technology, Japan
Shaoning Pang	Unitec Institute of Technology, New Zealand
Chi-Sing Leung	City University of Hong Kong, China

Registration Chair

Hai Zhao	Shanghai Jiao Tong University, China

Financial Chair

Yang Yang	Shanghai Maritime University, China

Local Arrangements Chairs

Guang Li	Zhejiang University, China
Fang Li	Shanghai Jiao Tong University, China

Secretary

Xun Liu Shanghai Jiao Tong University, China

Program Committee

Shigeo Abe
Bruno Apolloni
Sabri Arik
Sang-Woo Ban
Jianting Cao
Jonathan Chan
Songcan Chen
Xilin Chen
Yen-Wei Chen
Yiqiang Chen
Siu-Yeung David Cho
Sung-Bae Cho
Seungjin Choi
Andrzej Cichocki
Jose Alfredo Ferreira Costa
Sergio Cruces
Ke-Lin Du
Simone Fiori
John Qiang Gan
Junbin Gao
Xiaorong Gao
Nistor Grozavu
Ping Guo
Qing-Long Han
Shan He
Akira Hirose
Jinglu Hu
Guang-Bin Huang
Kaizhu Huang
Amir Hussain
Danchi Jiang
Tianzi Jiang
Tani Jun
Joarder Kamruzzaman
Shunshoku Kanae
Okyay Kaynak
John Keane
Sungshin Kim
Li-Wei Ko
Takio Kurita
Minho Lee
Chi Sing Leung
Chunshien Li
Guo-Zheng Li
Junhua Li
Wujun Li
Yuanqing Li
Yun Li
Huicheng Lian
Peiji Liang
Chin-Teng Lin
Hsuan-Tien Lin
Hongtao Lu
Libo Ma
Malik Magdon-Ismail
Robert(Bob) McKay
Duoqian Miao
Jun Miao
Vinh Nguyen
Tohru Nitta
Toshiaki Omori
Hassab Elgawi Osman
Seiichi Ozawa
Paul Pang
Hyeyoung Park
Alain Rakotomamonjy
Sarker Ruhul
Naoyuki Sato
Lichen Shi
Jochen J. Steil
John Sum
Jun Sun
Toshihisa Tanaka
Huajin Tang
Maolin Tang
Dacheng Tao
Qing Tao
Peter Tino

Ivor Tsang
Michel Verleysen
Bin Wang
Rubin Wang
Xiao-Long Wang
Yimin Wen
Young-Gul Won
Yao Xin
Rui-Feng Xu
Haixuan Yang
Jie Yang
Yang Yang
Yingjie Yang
Zhirong Yang
Dit-Yan Yeung
Jian Yu
Zhigang Zeng
Jie Zhang
Kun Zhang
Hai Zhao
Zhihua Zhou

Reviewers

Pablo Aguilera
Lifeng Ai
Elliot Anshelevich
Bruno Apolloni
Sansanee Auephanwiriyakul
Hongliang Bai
Rakesh Kr Bajaj
Tao Ban
Gang Bao
Simone Bassis
Anna Belardinelli
Yoshua Bengio
Sergei Bezobrazov
Yinzhou Bi
Alberto Borghese
Tony Brabazon
Guenael Cabanes
Faicel Chamroukhi
Feng-Tse Chan
Hong Chang
Liang Chang
Aaron Chen
Caikou Chen
Huangqiong Chen
Huanhuan Chen
Kejia Chen
Lei Chen
Qingcai Chen
Yin-Ju Chen
Yuepeng Chen
Jian Cheng
Wei-Chen Cheng
Yu Cheng
Seong-Pyo Cheon
Minkook Cho
Heeyoul Choi
Yong-Sun Choi
Shihchieh Chou
Angelo Ciaramella
Sanmay Das
Satchidananda Dehuri
Ivan Duran Diaz
Tom Diethe
Ke Ding
Lijuan Duan
Chunjiang Duanmu
Sergio Escalera
Aiming Feng
Remi Flamary
Gustavo Fontoura
Zhenyong Fu
Zhouyu Fu
Xiaohua Ge
Alexander Gepperth
M. Mohamad Ghassany
Adilson Gonzaga
Alexandre Gravier
Jianfeng Gu
Lei Gu
Zhong-Lei Gu
Naiyang Guan
Pedro Antonio Gutiérrez
Jing-Yu Han
Xianhua Han
Ross Hayward
Hanlin He
Akinori Hidaka
Hiroshi Higashi
Arie Hiroaki
Eckhard Hitzer
Gray Ho
Kevin Ho
Xia Hua
Mao Lin Huang
Qinghua Huang
Sheng-Jun Huang
Tan Ah Hwee
Kim Min Hyeok
Teijiro Isokawa
Wei Ji
Zheng Ji
Caiyan Jia
Nanlin Jin
Liping Jing
Yoonseop Kang
Chul Su Kim
Kyung-Joong Kim
Saehoon Kim
Yong-Deok Kim

Irwin King
Jun Kitazono
Masaki Kobayashi
Yasuaki Kuroe
Hiroaki Kurokawa
Chee Keong Kwoh
James Kwok
Lazhar Labiod
Darong Lai
Yuan Lan
Kittichai Lavangnananda
John Lee
Maylor Leung
Peter Lewis
Fuxin Li
Gang Li
Hualiang Li
Jie Li
Ming Li
Sujian Li
Xiaosong Li
Yu-feng Li
Yujian Li
Sheng-Fu Liang
Shu-Hsien Liao
Chee Peng Lim
Bingquan Liu
Caihui Liu
Jun Liu
Xuying Liu
Zhiyong Liu
Hung-Yi Lo
Huma Lodhi
Gabriele Lombardi
Qiang Lu
Cuiju Luan
Abdelouahid Lyhyaoui
Bingpeng Ma
Zhiguo Ma
Laurens Van Der Maaten
Singo Mabu
Shue-Kwan Mak
Asawin Meechai
Limin Meng
Komatsu Misako
Alberto Moraglio
Morten Morup
Mohamed Nadif
Kenji Nagata
Quang Long Ngo
Phuong Nguyen
Dan Nie
Kenji Nishida
Chakarida Nukoolkit
Robert Oates
Takehiko Ogawa
Zeynep Orman
Jonathan Ortigosa-Hernandez
Mourad Oussalah
Takashi J. Ozaki
Neyir Ozcan
Pan Pan
Paul S. Pang
Shaoning Pang
Seong-Bae Park
Sunho Park
Sakrapee Paul
Helton Maia Peixoto
Yong Peng
Jonas Peters
Somnuk Phon-Amnuaisuk
J.A. Fernandez Del Pozo
Santitham Prom-on
Lishan Qiao
Yuanhua Qiao
Laiyun Qing
Yihong Qiu
Shah Atiqur Rahman
Alain Rakotomamonjy
Leon Reznik
Nicoleta Rogovschi
Alfonso E. Romero
Fabrice Rossi
Gain Paolo Rossi
Alessandro Rozza
Tomasz Rutkowski
Nishimoto Ryunosuke
Murat Saglam
Treenut Saithong
Chunwei Seah
Lei Shi
Katsunari Shibata
A. Soltoggio
Bo Song
Guozhi Song
Lei Song
Ong Yew Soon
Liang Sun
Yoshinori Takei
Xiaoyang Tan
Chaoying Tang
Lei Tang
Le-Tian Tao
Jon Timmis
Yohei Tomita
Ming-Feng Tsai
George Tsatsaronis
Grigorios Tsoumakas
Thomas Villmann
Deng Wang
Frank Wang
Jia Wang
Jing Wang
Jinlong Wang
Lei Wang
Lu Wang
Ronglong Wang
Shitong Wang
Shuo Wang
Weihua Wang
Weiqiang Wang
Xiaohua Wang
Xiaolin Wang
Yuanlong Wang
Yunyun Wang
Zhikun Wang
Yoshikazu Washizawa
Bi Wei
Kong Wei
Yodchanan Wongsawat
Ailong Wu
Jiagao Wu

Jianxin Wu
Qiang Wu
Si Wu
Wei Wu
Wen Wu
Bin Xia
Chen Xie
Zhihua Xiong
Bingxin Xu
Weizhi Xu
Yang Xu
Xiaobing Xue
Dong Yang
Wei Yang
Wenjie Yang
Zi-Jiang Yang
Tianfang Yao
Nguwi Yok Yen
Florian Yger
Chen Yiming
Jie Yin
Lijun Yin
Xucheng Yin
Xuesong Yin
Jiho Yoo
Washizawa Yoshikazu
Motohide Yoshimura
Hongbin Yu
Qiao Yu
Weiwei Yu
Ying Yu
Jeong-Min Yun
Zeratul Mohd Yusoh
Yiteng Zhai
Biaobiao Zhang
Danke Zhang
Dawei Zhang
Junping Zhang
Kai Zhang
Lei Zhang
Liming Zhang
Liqing Zhang
Lumin Zhang
Puming Zhang
Qing Zhang
Rui Zhang
Tao Zhang
Tengfei Zhang
Wenhao Zhang
Xianming Zhang
Yu Zhang
Zehua Zhang
Zhifei Zhang
Jiayuan Zhao
Liang Zhao
Qi Zhao
Qibin Zhao
Xu Zhao
Haitao Zheng
Guoqiang Zhong
Wenliang Zhong
Dong-Zhuo Zhou
Guoxu Zhou
Hongming Zhou
Rong Zhou
Tianyi Zhou
Xiuling Zhou
Wenjun Zhu
Zhanxing Zhu
Fernando José Von Zube

Table of Contents – Part II

Cybersecurity and Data Mining Workshop

Agent Personalized Call Center Traffic Prediction and Call Distribution 1
Rafiq A. Mohammed and Paul Pang

Mapping from Student Domain into Website Category 11
Xiaosong Li

Entropy Based Discriminators for P2P Teletraffic Characterization 18
Tao Ban, Shanqing Guo, Masashi Eto, Daisuke Inoue, and Koji Nakao

Faster Log Analysis and Integration of Security Incidents Using Knuth-Bendix Completion 28
Ruo Ando and Shinsuke Miwa

Fast Protocol Recognition by Network Packet Inspection 37
Chuantong Chen, Fengyu Wang, Fengbo Lin, Shanqing Guo, and Bin Gong

Network Flow Classification Based on the Rhythm of Packets 45
Liangxiong Li, Fengyu Wang, Tao Ban, Shanqing Guo, and Bin Gong

Data Mining and Knowledge Discovery

Energy-Based Feature Selection and Its Ensemble Version 53
Yun Li and Su-Yan Gao

The Rough Set-Based Algorithm for Two Steps 63
Shu-Hsien Liao, Yin-Ju Chen, and Shiu-Hwei Ho

An Infinite Mixture of Inverted Dirichlet Distributions 71
Taoufik Bdiri and Nizar Bouguila

Multi-Label Weighted k-Nearest Neighbor Classifier with Adaptive Weight Estimation 79
Jianhua Xu

Emotiono: An Ontology with Rule-Based Reasoning for Emotion Recognition 89
Xiaowei Zhang, Bin Hu, Philip Moore, Jing Chen, and Lin Zhou

Parallel Rough Set: Dimensionality Reduction and Feature Discovery of Multi-dimensional Data in Visualization 99
Tze-Haw Huang, Mao Lin Huang, and Jesse S. Jin

Feature Extraction via Balanced Average Neighborhood Margin Maximization 109
Xiaoming Chen, Wanquan Liu, Jianhuang Lai, and Ke Fan

The Relationship between the Newborn Rats' Hypoxic-Ischemic Brain Damage and Heart Beat Interval Information 117
Xiaomin Jiang, Hiroki Tamura, Koichi Tanno, Li Yang, Hiroshi Sameshima, and Tsuyomu Ikenoue

A Robust Approach for Multivariate Binary Vectors Clustering and Feature Selection 125
Mohamed Al Mashrgy, Nizar Bouguila, and Khalid Daoudi

The Self-Organizing Map Tree (SOMT) for Nonlinear Data Causality Prediction 133
Younjin Chung and Masahiro Takatsuka

Document Classification on Relevance: A Study on Eye Gaze Patterns for Reading 143
Daniel Fahey, Tom Gedeon, and Dingyun Zhu

Multi-Task Low-Rank Metric Learning Based on Common Subspace 151
Peipei Yang, Kaizhu Huang, and Cheng-Lin Liu

Reservoir-Based Evolving Spiking Neural Network for Spatio-temporal Pattern Recognition 160
Stefan Schliebs, Haza Nuzly Abdull Hamed, and Nikola Kasabov

An Adaptive Approach to Chinese Semantic Advertising 169
Jin-Yuan Chen, Hai-Tao Zheng, Yong Jiang, and Shu-Tao Xia

A Lightweight Ontology Learning Method for Chinese Government Documents 177
Xing Zhao, Hai-Tao Zheng, Yong Jiang, and Shu-Tao Xia

Relative Association Rules Based on Rough Set Theory 185
Shu-Hsien Liao, Yin-Ju Chen, and Shiu-Hwei Ho

Scalable Data Clustering: A Sammon's Projection Based Technique for Merging GSOMs 193
Hiran Ganegedara and Damminda Alahakoon

A Generalized Subspace Projection Approach for Sparse Representation Classification 203
Bingxin Xu and Ping Guo

Evolutionary Design and Optimisation

Macro Features Based Text Categorization 211
Dandan Wang, Qingcai Chen, Xiaolong Wang, and Buzhou Tang

Univariate Marginal Distribution Algorithm in Combination with Extremal Optimization (EO, GEO) 220
Mitra Hashemi and Mohammad Reza Meybodi

Promoting Diversity in Particle Swarm Optimization to Solve Multimodal Problems 228
Shi Cheng, Yuhui Shi, and Quande Qin

Analysis of Feature Weighting Methods Based on Feature Ranking Methods for Classification 238
Norbert Jankowski and Krzysztof Usowicz

Simultaneous Learning of Instantaneous and Time-Delayed Genetic Interactions Using Novel Information Theoretic Scoring Technique 248
Nizamul Morshed, Madhu Chetty, and Nguyen Xuan Vinh

Resource Allocation and Scheduling of Multiple Composite Web Services in Cloud Computing Using Cooperative Coevolution Genetic Algorithm 258
Lifeng Ai, Maolin Tang, and Colin Fidge

Graphical Models

Image Classification Based on Weighted Topics 268
Yunqiang Liu and Vicent Caselles

A Variational Statistical Framework for Object Detection 276
Wentao Fan, Nizar Bouguila, and Djemel Ziou

Performances Evaluation of GMM-UBM and GMM-SVM for Speaker Recognition in Realistic World 284
Nassim Asbai, Abderrahmane Amrouche, and Mohamed Debyeche

SVM and Greedy GMM Applied on Target Identification 292
Dalila Yessad, Abderrahmane Amrouche, and Mohamed Debyeche

Speaker Identification Using Discriminative Learning of Large Margin GMM 300
Khalid Daoudi, Reda Jourani, Régine André-Obrecht, and Driss Aboutajdine

Sparse Coding Image Denoising Based on Saliency Map Weight 308
Haohua Zhao and Liqing Zhang

Human-Originated Data Analysis and Implementation

Expanding Knowledge Source with Ontology Alignment for Augmented Cognition 316
Jeong-Woo Son, Seongtaek Kim, Seong-Bae Park, Yunseok Noh, and Jun-Ho Go

Nyström Approximations for Scalable Face Recognition: A Comparative Study 325
Jeong-Min Yun and Seungjin Choi

A Robust Face Recognition through Statistical Learning of Local Features 335
Jeongin Seo and Hyeyoung Park

Development of Visualizing Earphone and Hearing Glasses for Human Augmented Cognition 342
Byunghun Hwang, Cheol-Su Kim, Hyung-Min Park, Yun-Jung Lee, Min-Young Kim, and Minho Lee

Facial Image Analysis Using Subspace Segregation Based on Class Information 350
Minkook Cho and Hyeyoung Park

An Online Human Activity Recognizer for Mobile Phones with Accelerometer 358
Yuki Maruno, Kenta Cho, Yuzo Okamoto, Hisao Setoguchi, and Kazushi Ikeda

Preprocessing of Independent Vector Analysis Using Feed-Forward Network for Robust Speech Recognition 366
Myungwoo Oh and Hyung-Min Park

Information Retrieval

Learning to Rank Documents Using Similarity Information between Objects 374
Di Zhou, Yuxin Ding, Qingzhen You, and Min Xiao

Efficient Semantic Kernel-Based Text Classification Using Matching Pursuit KFDA 382
Qing Zhang, Jianwu Li, and Zhiping Zhang

Introducing a Novel Data Management Approach for Distributed Large Scale Data Processing in Future Computer Clouds 391
Amir H. Basirat and Asad I. Khan

PatentRank: An Ontology-Based Approach to Patent Search 399
Ming Li, Hai-Tao Zheng, Yong Jiang, and Shu-Tao Xia

Fast Growing Self Organizing Map for Text Clustering 406
Sumith Matharage, Damminda Alahakoon, Jayantha Rajapakse, and Pin Huang

News Thread Extraction Based on Topical N-Gram Model with a Background Distribution 416
Zehua Yan and Fang Li

Integrating Multiple Nature-Inspired Approaches

Alleviate the Hypervolume Degeneration Problem of NSGA-II 425
Fei Peng and Ke Tang

A Hybrid Dynamic Multi-objective Immune Optimization Algorithm Using Prediction Strategy and Improved Differential Evolution Crossover Operator 435
Yajuan Ma, Ruochen Liu, and Ronghua Shang

Optimizing Interval Multi-objective Problems Using IEAs with Preference Direction 445
Jing Sun, Dunwei Gong, and Xiaoyan Sun

Fitness Landscape-Based Parameter Tuning Method for Evolutionary Algorithms for Computing Unique Input Output Sequences 453
Jinlong Li, Guanzhou Lu, and Xin Yao

Introducing the Mallows Model on Estimation of Distribution Algorithms 461
Josu Ceberio, Alexander Mendiburu, and Jose A. Lozano

Kernel Methods and Support Vector Machines

Support Vector Machines with Weighted Regularization 471
Tatsuya Yokota and Yukihiko Yamashita

Relational Extensions of Learning Vector Quantization 481
Barbara Hammer, Frank-Michael Schleif, and Xibin Zhu

On Low-Rank Regularized Least Squares for Scalable Nonlinear Classification 490
Zhouyu Fu, Guojun Lu, Kai-Ming Ting, and Dengsheng Zhang

Multitask Learning Using Regularized Multiple Kernel Learning 500
Mehmet Gönen, Melih Kandemir, and Samuel Kaski

Solving Support Vector Machines beyond Dual Programming 510
Xun Liang

Learning with Box Kernels 519
Stefano Melacci and Marco Gori

A Novel Parameter Refinement Approach to One Class Support Vector Machine 529
Trung Le, Dat Tran, Wanli Ma, and Dharmendra Sharma

Multi-Sphere Support Vector Clustering 537
Trung Le, Dat Tran, Phuoc Nguyen, Wanli Ma, and Dharmendra Sharma

Testing Predictive Properties of Efficient Coding Models with Synthetic Signals Modulated in Frequency 545
Fausto Lucena, Mauricio Kugler, Allan Kardec Barros, and Noboru Ohnishi

Learning and Memory

A Novel Neural Network for Solving Singular Nonlinear Convex Optimization Problems 554
Lijun Liu, Rendong Ge, and Pengyuan Gao

An Extended TopoART Network for the Stable On-line Learning of Regression Functions 562
Marko Tscherepanow

Introducing Reordering Algorithms to Classic Well-Known Ensembles to Improve Their Performance 572
Joaquín Torres-Sospedra, Carlos Hernández-Espinosa, and Mercedes Fernández-Redondo

Improving Boosting Methods by Generating Specific Training and Validation Sets 580
Joaquín Torres-Sospedra, Carlos Hernández-Espinosa, and Mercedes Fernández-Redondo

Using *Bagging* and *Cross-Validation* to Improve Ensembles Based on Penalty Terms 588
Joaquín Torres-Sospedra, Carlos Hernández-Espinosa, and Mercedes Fernández-Redondo

A New Algorithm for Learning Mahalanobis Discriminant Functions by a Neural Network 596
Yoshifusa Ito, Hiroyuki Izumi, and Cidambi Srinivasan

Learning of Dynamic BNN toward Storing-and-Stabilizing Periodic Patterns 606
Ryo Ito, Yuta Nakayama, and Toshimichi Saito

Self-organizing Digital Spike Interval Maps 612
Takashi Ogawa and Toshimichi Saito

Shape Space Estimation by SOM^2 618
Sho Yakushiji and Tetsuo Furukawa

Neocognitron Trained by Winner-Kill-Loser with Triple Threshold 628
Kunihiko Fukushima, Isao Hayashi, and Jasmin Léveillé

Nonlinear Nearest Subspace Classifier 638
Li Zhang, Wei-Da Zhou, and Bing Liu

A Novel Framework Based on Trace Norm Minimization for Audio Event Detection 646
Ziqiang Shi, Jiqing Han, and Tieran Zheng

A Modified Multiplicative Update Algorithm for Euclidean Distance-Based Nonnegative Matrix Factorization and Its Global Convergence 655
Ryota Hibi and Norikazu Takahashi

A Two Stage Algorithm for K-Mode Convolutive Nonnegative Tucker Decomposition 663
Qiang Wu, Liqing Zhang, and Andrzej Cichocki

Making Image to Class Distance Comparable 671
Deyuan Zhang, Bingquan Liu, Chengjie Sun, and Xiaolong Wang

Margin Preserving Projection for Image Set Based Face Recognition 681
Ke Fan, Wanquan Liu, Senjian An, and Xiaoming Chen

An Incremental Class Boundary Preserving Hypersphere Classifier 690
Noel Lopes and Bernardete Ribeiro

Co-clustering for Binary Data with Maximum Modularity 700
Lazhar Labiod and Mohamed Nadif

Co-clustering under Nonnegative Matrix Tri-Factorization 709
Lazhar Labiod and Mohamed Nadif

SPAN: A Neuron for Precise-Time Spike Pattern Association 718
Ammar Mohemmed, Stefan Schliebs, and Nikola Kasabov

Induction of the Common-Sense Hierarchies in Lexical Data 726
Julian Szymański and Włodzisław Duch

A Novel Synthetic Minority Oversampling Technique for Imbalanced Data Set Learning 735
Sukarna Barua, Md. Monirul Islam, and Kazuyuki Murase

A New Simultaneous Two-Levels Coclustering Algorithm for Behavioural Data-Mining 745
Guénaël Cabanes, Younès Bennani, and Dominique Fresneau

An Evolutionary Fuzzy Clustering with Minkowski Distances 753
Vivek Srivastava, Bipin K. Tripathi, and Vinay K. Pathak

A Dynamic Unsupervised Laterally Connected Neural Network Architecture for Integrative Pattern Discovery 761
Asanka Fonseka, Damminda Alahakoon, and Jayantha Rajapakse

Author Index 771

Agent Personalized Call Center Traffic Prediction and Call Distribution

Rafiq A. Mohammed and Paul Pang

Department of Computing, Unitec Institute of Technology,
Private Bag 92025, New Zealand
ppang@unitec.ac.nz

Abstract. A call center operates with customers calls directed to agents for service based on online call traffic prediction. Existing methods for call prediction implement exclusively inductive machine learning, which gives often under accurate prediction for call center abnormal traffic jam. This paper proposes an agent personalized call prediction method that encodes agent skill information as the prior knowledge to call prediction and distribution. The developed call broker system is tested on handling a telecom call center traffic jam happened in 2008. The results show that the proposed method predicts the occurrence of traffic jam earlier than existing depersonalized call prediction methods. The conducted cost-return calculation indicates that the ROI (return on investment) is enormously positive for any call center to implement such an agent personalized call broker system.

Keywords: Call Center Management, Call Traffic Prediction, Call Traffic Jam, Agent Personalized Call Broker.

1 Introduction

Call centers are the backbone of any service industry. A recent McKin-sey study revealed that for credit card companies generate up to 25% of new revenue from inbound calls center's [13]. The telecommunication industry is improving at a very high speed [8], the total number of mobile phone users has exceeded 400 million by September 2006; and this immense market growth has generated a cutthroat competition among the service providers. These scenarios have brought up the need of call center's, which can offer quality services over the phone to stay in a competitive environment. A call center handles calls from several queues and mainly consists of residential, mobile, business and broadband customers. The faults call center queues operates 24 hours a day and 7 days a week.

Fig. 1 gives the diagram of call center call processing. The Interactive Voice Response (IVR) system, initially takes up the call from the customer, and performs a diagnosis conversation of the problem with the customer, such that the problem can be resolved on-line with the process of self-check with the customer. If the problem is not resolved, the system will divert the call to the software broker, which actually understands the problem by looking at the paraphrased

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 1–10, 2011.
© Springer-Verlag Berlin Heidelberg 2011

problem description. The broker is going to request for a list of available agents to a search engine, such that it can link the path of the call to an agent queue with the help of Automatic Call Distributor (ACD). From the available list, the broker requests supervisor to assist in selection criteria. The supervisor monitors agent performance from the agent and customer databases (DB) and evaluates when it is required to select a better agent for a customer in queue. The search engine list the most appropriate agent based on agent database and supervisor recommendations.

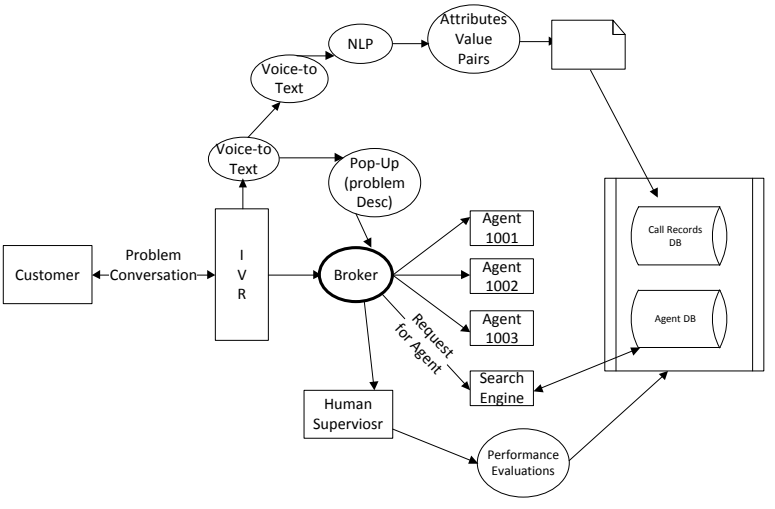


Fig. 1. Diagram of call center call processing

For the broker system in Fig. 1, calls are routed based on the availability of the skilled agent for which the call was made for. If the primary skilled agent is not available, the call will be routed to the secondary skilled agent. However, if 'm' number of primary skilled or 'n', number of secondary skilled agents are available to answer the calls; the ACD will allocate the calls giving priority to the agents who have been waiting for the longest time. Obviously, this is not an efficient call broke approach because the skill of each agent actually is different from one another. Nevertheless, such call broker systems have been widely used in call centers for automatic call distribution and they are working well to handle normal traffic call flows.

However, traffic jam occurs sometime in a call center, even if the above call broker systems are used. While traffic jam, the call arrival pattern displays for a certain period an unusual size of call volume per day, as well as an abnormal call distribution. Analyzing the facts for these unusual call distributions, it is found that they are often caused by some technical accident, for example a major telecom exchange system was down and it caused an increase in the number of calls coming to the call center. This paper studies a new IT solution to such call

center traffic jam, and proposes an agent personalized call prediction and call distribution model.

2 Call Center Management

A call center is a dedicated operation with employees focus entirely on offering customer service [1]. While performing business tasks there raise a question, how can we perform trade-off between customer service quality (CSQ) and effciency of business operations (EBO)? A better customer service will bring benefits for customers such as service quality [4], satisfaction [2, 3] for efficient resolutions of their problems. This will in-turn generate customer loyalty, effective business solutions, revenue producer and competitive market share for organization and finally bring a sort of job satisfaction to the agents for offering efficient customer solutions.

2.1 CSQ Measurements

Telephone Service Factor (TSF) is a quality measure in a call center, which tells us the percentage of incoming calls answered or abandoned within the customer defined threshold time. The quickness of calls answered or abandoned would be a usual measure of TSF. The customer specifies the time (in seconds) in the programming of the telephone system. The usual result would be a percentage of calls that falls with in that threshold time.

AverageWork Time (AWT) measures the efficiency of agent performance in a call center. AWT is computed as (Login time-wait time)/(No of calls Answered). Login time denotes the state, in which agents have signed on to a system to make their presence known, but may or may not be ready to receive calls. Wait time denotes the availability of agents to receive calls. For example, Telecom NewZealand (TNZ) assures AWT of 6 minutes as an effective benchmark to calculate agent's efficiency. In addition to TSF and AWT, other CSQ measurements include Average Speed of Answer (ASA) [8], Call Abundances (CA), Recall/First Call Resolution (FCR) [8], and Average Handling Time (AHT).

2.2 EBO and Trade-Off between CSQ and EBO

On the other side, call center measures efficiency of business operations based on a) staff efficiency and b) cost efficiency. Bringing out some of the approaches of organizations, an airline industry has chosen to allow some loss of service to the customer reservations system such that they can save large costs of staffing during heavy traffic periods and thus deviated TSF norm deliberately in favor of economic considerations.

Looking at aspects for resolving trade-off between CSQ and EBO, organizations attempt to meet both monetary and service priorities and are often leading to confflicts such as "hard versus soft goals", "intangible versus tangible outcomes" [3], "Taylorism versus tailorism" while managing call center. The

organization has to maintain a balance between customer service quality and efficiency of business operations, as loss of service to efficiency can influence its future. In [4], this idea has been supported that who perceived customer loyalty to the organization has a positive relation with service quality of the call center. The call center is no more a cost center, as a good customer service generates loyalty and revenue to the organization. Many businesses are coming out of dilemma to consider call centers as a strategic revenue generating units rather than purely as a cost center while offering customer service [2].

3 Review of Call-Center IT Solutions

According to [2], researchers develop several types of optimization, queuing and simulations models, heuristics and algorithms to help decrease customer wait times, increase throughput, and increase customer satisfaction. Such research efforts have led to several real-time scheduling techniques and optimization models that enable call center to manage capacity more efficiently, even when faced with highly fluctuating demand.

Erlang C: The Erlang-C queuing model M/M/n assumes calls arrive at Poisson arrival rate, the service time is exponentially distributed and there are n agents with identical statistical details. However, it is deficient as an accurate depiction of a call center in some major respects: it does not include priorities of customers; it assumes that skills of agents and their service-time distributions are identical, it ignores customers' recalls, etc. [5] and ignores call abandonment's as well.

Erlang A: Bringing in the defects of Erlang C model for ignoring call abandonment's [6] analyzed the simplest abandonment model M/M/n+M (Erlang-A), in which customers' patience is exponentially distributed; such that customer satisfaction and call abandonment's are calculated. In addition, "Rules of thumb" for the design and staffing of medium to large call centers were then derived [5].

Erlang B: It is widely used to determine the number of trunks required to handle a known calling load during a one-hour period. The formula assumes that if callers get busy signals, they go away forever, never to retry (lost calls cleared). Since some callers retry, Erlang B can underestimate trunks required. However, Erlang B is generally accurate in situations with few busy signals as it incorporates blocking of customers.

Telecommunication call center often uses the queuing model like Erlang A & Erlang C for the operations of optimization [8]. As observed from the TNZ case study, the call center uses Erlang C to predict agent requirements based on forecasted call volumes & handle times with the use of excel spreadsheets. TNZ also use a workforce management tool called "ResourcePro" that does the scheduling of agents.

Data Warehousing (DWH): Looking at the works of researcher [8] use of OLAP (On-Line Analytical Processing) and data mining manage to mine service quality metrics such as ASA, recall, IVR optimization to improve the service quality. However, if we include agent DB with in the DWH, it is possible to monitor and evaluate the performance of agents, such that to improve call quality and customer service satisfaction.

Data Mining: Looking at some of the advancements of CIT, with Predictive modelling such as decision-tree or neural network based techniques it is possible to predict customer behavior, and the analysis of customer behavior with data mining aims to improve customer satisfaction [7].

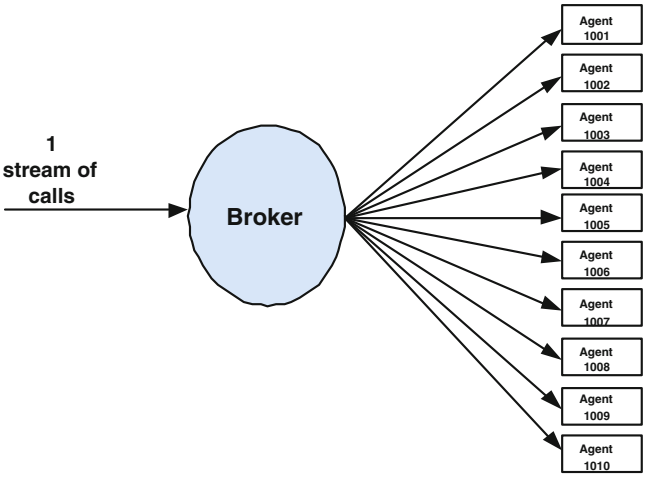


Fig. 2. Diagram of call center broker system with depersonalized call prediction model

4 Existing Call Prediction Methods

In literature, several inductive machine learning methods has been investigated and used for call volume prediction of call center. This includes, (1) DENFIS (Dynamic Evolving Neural-Fuzzy Inference System), a method of fuzzy interface systems for online and/or offline adaptive learning [9]. DENFIS adapts new features from the dynamic change of data, thus is capable of predicting the dynamic time series prediction efficiently; (2) MLR (Multiple Linear regressions), a statistical multivariate least squares regression method. This method takes a dataset with one output but multiple input variables, seeking a formula that approximates the data samples that can in linear regression. The obtained regression formula is used as a prediction model for any new input vectors; (3) MLP (Multilayer Perceptrons), a standard neural network model for learning from data as a non-linear function that discriminates (or approximates) data according to output labels (values). Additionally, it is worth noting that the experience-based prediction is popularly used for call prediction. Such methods use an estimator drawn from past experience and expectations to forecast future call traffic parameters. Fig. 2 illustrates the scenario of de-personalized broker, where the stream of calls is allocated by an automatic call distributor (broker) to the available agents irrespective of the skills of the agents. In other words, call is equally distributed to agents, regardless of the skill differences amongst agents. In practice, such de-personalized model could be suitable for a call center of 5-6 agents. However for a call center

greater than 50 agents, such de-personalized call prediction/distribution actually deducts the efficiency of business operations, as well as the customer service quality of the call center. In the scenario of handling large number of agents, an alternative approach is to introduce an agent personalized call prediction method at the automatic call distributor (broker) software system.

5 Proposed Call Prediction Method

The idea of personalized call broker is from Fig. 3. With agent personalized call prediction, the broker system works virtually having a number of brokers personalized to each agent, rather than a single generalist broker for all the agents. This makes the problem simpler to predict the appropriate calls to the each individual agent of the whole agent team [10]. Implementing such system at ACD is expected to improve the functionality of broker and will bring us real time approaches to automatic call distribution.

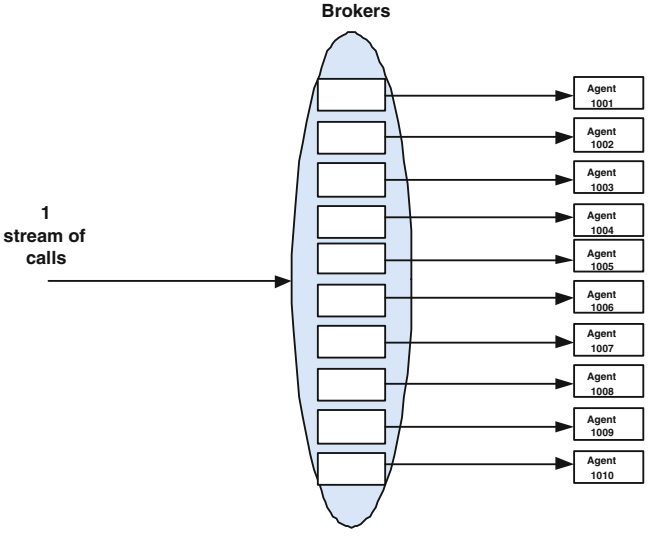


Fig. 3. Diagram of agent personalized call center broker system

5.1 Agent Personalized Prediction

Assume that a call center has total m agents, traditional broker system maintains as in Fig. 2 one general call volume prediction, and distribute calls equally to m agents. Obviously, this is not an efficient approach as the skill of each agent is different from one another.

Given a data stream $D = \{y(i), y(i+1), \ldots, y(i+t)\}$, representing a certain period of historical call volume confronted by the call center, depersonalized call prediction can be formulated as,

$$y(i+t+1) = f(y(i), y(i+1), \ldots, y(i+t)) \tag{1}$$

where $y(i)$ represents the number of calls at a certain time point i, f is the base prediction function, which could be a Multivariate Linear Regressions (MLR), Neural Network, or any other type of prediction method described above.

Introducing the skill grade of each agent $S = \{s_1, s_2, \ldots, s_m\}$ as the prior knowledge to the predictor, we have the call volume decomposed into m data streams accordingly. Then, the number of call on each agent is calculated as,

$$z^{(j)}(t) = \frac{y(t)s_j}{\sum_{i=1} S_i}, j \in [1; m]. \tag{2}$$

Partitioning data stream D by (2) and applying (1) to each individual data stream obtained from (2), we have,

$$\begin{array}{rcl} z^{(1)}(i+t+1) & = & f(1)(z^{(1)}(i), z^{(1)}(i+1), \ldots, z^{(1)}(i+t)) \\ z^{(2)}(i+t+1) & = & f(2)(z^{(2)}(i), z^{(2)}(i+1), \ldots, z^{(2)}(i+t))) \\ \ldots \quad \ldots & & \ldots \\ z^{(m)}(i+t+1) & = & f(m)(z^{(m)}(i), z^{(m)}(i+1), \ldots, z^{(m)}(i+t)). \end{array} \tag{3}$$

Since $y(t+1) = \sum_{j=1}^{m} z(j)(t+1)$, a personalized prediction model for call traffic prediction can be formulated as,

$$\begin{array}{l} y(i+t+1) = \Omega(f^{(1)}, f^{(2)}, \ldots, f^{(m)}, S) \\ \qquad\qquad = \frac{1}{m} \sum_{j=1}^{m} z^{(j)}(i+t+1) \end{array} \tag{4}$$

where Ω is the personalized prediction model based on the prior knowledge from agent skill information.

6 Experiments and Discussion

The datasets originated from a New Zealand telecommunication industry call center. The call data consists of detailed call-by-call histories obtained from the faults resolve department. The call data to the system arrives regularly at 15 minutes intervals and for the entire day. The queues are busy mostly between the operating hours of 7 AM and 11 PM. In order to bring a legitimate comparison, data from 07:00 to 23:00 hours will be considered for data analysis and practical investigation. For traffic jam call prediction, the dataset consists of 40 days of call volume data between dates of 22/01/2008 till 01/03/2008. The first 30 days have a normal distribution and the last 10 days depict a traffic jam. A sliding window approach is implemented to predict the next day's call volume, whereby for each subsequent day of prediction the window will be moved one day ahead. This approach will predict the call volume for 10 days of traffic jam period. To exhibit the advantages of our method, we use a standard MLR as the base prediction function, and evaluate prediction performance by both call volume in terms of the number of calls, and the root mean squared error (RMSE).

Table 1. Call volume in terms of the number of calls, and the root mean squared error (RMSE)

Methods	Tr (days)	Tp (days)	St (days)	Cost Saving (%)
De-personalized	3.60	8.60	1.40	(52,700-38,419)/52,700=27%
Personalized	3.48	8.487	1.52	(52,700-45,308)/52,700=14%

6.1 Traffic Jam Call Prediction

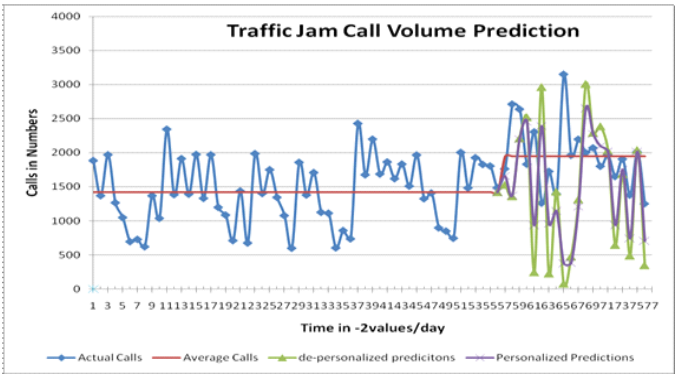
Fig. 4 gives a comparison between the proposed agent personalized method versus the depersonalized prediction method for call forecasting within the period of traffic jam. As seen from the experimental results, utilizing agent skills as the prior knowledge to personalized prediction gives us a superior call volume prediction accuracy and lower RMSE than the typical prediction method. Assuming that the 10 days traffic jam follows generally a Gaussian distribution, then the traffic jam reaches its peak on the 5th day, which is the midpoint of the traffic jam period. Consider 5 days as the constant parameter, we calculate the predicted traffic jam period as $Tp = Ts + Tr$. Here, Ts is the starting point of traffic jam, which is normally determined by, if current 5-day average traffic volume is greater than the threshold of traffic jam average daily call volume. Tr is the time to release the traffic jam calculated by $(A - N = P)$, where A is actual calls during the traffic jam period; N is calls for the period in the case of normal traffic; and P is average daily predicted calls by each method during traffic jam period. The time saving due to call prediction St is calculated by subtracting the total time of prediction from traffic jam period, which is $10 - Tp$.

Table 1 presents the traffic jam release time and time savings due to call prediction. It is evident that personalized call predictions save us 1.52 days, which is better than the 1.40 days from typical de-personalized call predictions.

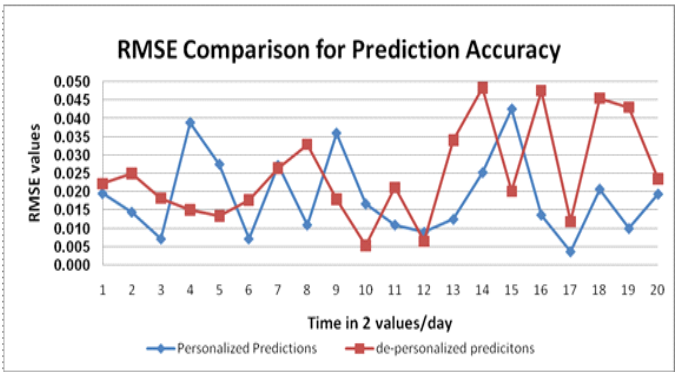
6.2 Cost & Return Evaluation

According to [11], the operating cost in a call center includes, agents salaries, network cost, and management cost, where agents salaries typically account for 60% to 70 % of the total operating costs. Considering an additional cost of $52,700 for the 10 days traffic jam, introducing traffic jam problem solving, the de-personalized call prediction release the traffic jam in 8.60 days with a total cost of $45,308. This is in contrast to the agent personalized prediction that releases the same traffic jam in 8.48 days with a total cost of $38,419 and a saving of 27%. Table. 1 records the traffic jam cost saving due to call prediction by different methods.

On the other hand, while computing the cost of single supervisor, it will incur an additional cost of $1151 for a 10-day period to hire a new supervisor to manage the call center. According to [12], the cost of hiring additional supervisor amounts to $42,000 per year to manage a call center. Thus from the cost and



(a)



(b)

Fig. 4. Comparisons of personalized versus de-personalized methods for traffic jam period call prediction, (a) call volume predictions and (b) root mean square error (RMSE)

return calculation, it is beneficial for any call center to implement personalized call broker model, as there is a minimum net saving of $20,666 as return on investment.

7 Conclusions

This paper develops a new call broker model that implements an agent personalized call prediction approach towards enhancing the call distribution capability of existing call broker. In our traffic jam problem investigation, the proposed personalized call broker model is demonstrated capable of releasing traffic jam earlier than the existing depersonalized system. Addressing telecommunication industry call center management, the presented research brings the awareness of call center traffic jam, appealing for change in call prediction models to foresee and avoid future call center traffic jams.

References

1. Taylor, P., Bain, P.: An assembly line in the head': work and employee relations in the call centre. Industrial Relations Journal 30(2), 101–117 (1999)
2. Jack, E.P., Bedics, T.A., McCary, C.E.: Operational challenges in the call center industry: a case study and resource-based framework. Managing Service Quality 16(5), 477–500 (2006)
3. Gilmore, A., Moreland, L.: Call centres: How can service quality be managed. Irish Marketing Review 13(1), 3–11 (2000)
4. Dean, A.M.: Service quality in call centres: implications for customer loyalty. Managing Service Quality 12(6), 414–423 (2002)
5. Mandelbaum, A., Zeltyn, S.: The impact of customers' patience on delay and abandonment: some empirically-driven experiments with the M/M/n+ G queue. OR Spectrum 26(3), 377–411 (2004)
6. Garnet, O., Mandelbaum, A., Reiman, M.: Designing a Call Center with Impatient Customers. Manufacturing and Service Operations Management 4(3), 208–227 (2002)
7. Paprzycki, M., Abraham, A., Guo, R.: Data Mining Approach for Analyzing Call Center Performance. Arxiv preprint cs.AI/0405017 (2004)
8. Shu-guang, H., Li, L., Er-shi, Q.: Study on the Continuous Quality Improvement of Telecommunication Call Centers Based on Data Mining. In: Proc. of International Conference on Service Systems and Service Management, pp. 1–5 (2007)
9. Kasabov, N.K., Song, Q.: DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Transactions on Fuzzy Systems 10(2), 144–154 (2002)
10. Pang, S., Ban, T., Kadobayashi, Y., Kasabov, N.: Tansductive Mode Personalized Support Vector Machine Classification Tree. Information Science 181(11), 2071–2085 (2010)
11. Gans, N., Koole, G., Mandelbaum, A.: Telephone call centers: Tutorial, review, and research prospects. Manufacturing and Service Operations Management 5(2), 79–141 (2003)
12. Hillmer, S., Hillmer, B., McRoberts, G.: The Real Costs of Turnover: Lessons from a Call Center. Human Resource Planning 27(3), 34–42 (2004)
13. Eichfeld, A., Morse, T.D., Scott, K.W.: Using Call Centers to Boost Revenue. McKinsey Quarterly (May 2006)

Mapping from Student Domain into Website Category

Xiaosong Li

Department of Computing and IT, Unitec Institute of Technology,
Auckland, New Zealand
xli@unitec.ac.nz

Abstract. The existing e-learning environments focus on the reusability of learning resources not adaptable to suit learners' needs [1]. Previous research shows that users' personalities have impact on their Internet behaviours and preferences. This study investigates the relationship between user attributes and their website preferences by using a practical case which suggested that there seemed to be relationships between a student's gender, age, mark and the type of the website he/she has chosen aiming to identify valuable information which can be utilised to provide adaptive e-learning environment for each student. This study builds ontology taxonomy in the student domain first, and then builds ontology taxonomy in the website category domain. Mapping probabilities are defined and used to generate the similarity measures between the two domains. This study uses two data sets. The second data set was used to learn similarity measures and the first data set was used to test the similarity formula. The scope of this study is not limited to e-learning system. The similar approach may be used to identify potential sources of Internet security issues.

Keywords: ontology, taxonomy, student, website category, Internet security.

1 Introduction

The existing e-learning environments are certainly helpful for student learning. However, they focus on the reusability of learning resources not adaptable to suit learners' needs [1]. To encourage student centred learning and help student actively engaged in the learning process, we need to promote student self-regulated learning (SRL), in which the learner has to use specified strategies for attaining his or her goals and all this has to be based self-efficacy perceptions. Learner oriented environments claim a greater extent of self-regulated learning skills [2]. A learner oriented e-learning environment should provide adaptive instruction, guideline and feedback to the specific learner. Creation of the educational Semantic Web provides more opportunity in this area [3].

Previous research shows that users' personalities have impact on their Internet behaviours and preferences [4]. For example, people have a need for closure are motivated to avoid uncertainties; people who are conformists are likely to prefer a website with many constant components and find it stressful if the website is changed frequently; whereas people who are innovators will be stimulated with a website that changes [4]. This study investigates the relationship between user attributes and their website preferences by using a practical case. In our first year website and multimedia

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 11–17, 2011.
© Springer-Verlag Berlin Heidelberg 2011

course, there is an assignment which requires the students to choose an existing website to critique. The selection is mutually exclusive, which means as long as a website has been chosen by a student, the others can not choose the same website anymore. After two semesters' observation, it was found that the websites chosen by the students have fallen into certain categories, such as news websites, sports websites and etc. And there seemed to be relationships between a student's gender, age, mark and the type of the website he/she had chosen. For example, all the students who had chosen a sports website are male students. This paper reports a pilot investigation on this phenomena aiming to identify valuable information which can be utilised to provide adaptive contents, services, instructions, guidelines and feedbacks in an e-learning environment for each student. For example, when a student input their personal data such as gender, age, and etc., the system could identify a couple of categories of website or the actual websites he/she might be interested.

This study builds ontology taxonomy on students' personal attributes in student domain first, and then builds ontology taxonomy in website category domain. Mapping probabilities are defined and used to generate the similarity measures between the two domains. This study uses two data sets. The second data set was used to learn similarity measures and the first data set was used to test the similarity formula. In the following sections, the data used in this study is described first, followed by ontology taxonomy definitions, and then similarity measures and their testing are described, finally the results and the possible future work are discussed.

2 The Data

The data from two semesters were used, each generated one data set. Initially, the categories for the chosen websites were identified. Only those categories selected by more than one student for both of the semesters were considered. Nine categories were identified. Table 1 shows the name, a short description and the given code for each category which will be used in the rest of the paper.

Table 1. The identified website categories

Name	Short Description	Code
Finance	Bank or finance organization	FI
Game	Game or game shopping	GA
Sports	Sports or sports related shopping	SP
Computing/IT	Computing technology or shopping	CO
Market	E Commerce or online store	MA
Knowledge	Library or Wikipedia ……	KN
Telecommunication	Telecommunication business	TE
Travel/Holiday	Travel or holiday related ……	TR
News	News related ……	NE

The students who didn't complete the assignment or the selected website was not active were eliminated from the data set. For the students who made multiple selections, their last selection was included in the data. As a result, each student is

associated with one and only one website in the data set. Each student is also associated with an assessment factor, which is their assignment mark divided by the average mark of that data set. The purpose of this is to minimize external factors which might impact the marks in a particular semester. Two data sets are presented; each is grouped according to the website category. Table 2 shows the summary of the first data set; and Table 3 second data set.

Table 2. The first data set. AV Age = average age in the group; Number = instance number in the group; M:F = Male : Female; AV Ass = average assessment factor in the group. The data size = 31.

Code	AV Age	M:F	AV Ass	Number
FI	27.50	3:1	1.13	4
GA	22.75	3:1	0.93	4
SP	24.67	3:0	1.01	3
CO	21.50	2:0	0.84	2
MA	19.50	3:0	1.01	3
KN	21.00	2:1	1.13	3
TE	22.33	4:0	0.85	4
TR	33.00	4:1	1.09	5
NE	24.33	1:2	0.93	3

Table 3. The second data set. AV Age = average age in the group; Number = instance number in the group; M:F = Male : Female; AV Ass = average assessment factor in the group. The data size = 32.

Code	AV Age	M:F	AV Ass	Number
FI	23.50	1:1	1.09	2
GA	20.00	4:1	1.00	5
SP	21.67	6:0	1.04	6
CO	21.60	3:2	1.02	5
MA	21.25	4:0	1.04	4
KN	20.00	2:0	0.89	2
TE	23.00	1:1	1.07	2
TR	24.75	2:2	1.07	4
NE	22.00	0:2	0.87	2

3 The Concepts

Two ontology taxonomies were built in this pilot study. One is **Student**, another is **Website Categories**.

Fig 1. shows the outlines of the two ontologies. The similar specification in [5] is adopted, where each node represents a concept which may be associated with a set of attributes or instances. The first ontology depicted in part A contains two concepts: **Student** and **Selected Website**. The attributes for **Student** include **name**, **gender**, **assessment factor** and **age**. The attribute for **Selected Website** include a **url**. The

instances of **Student** are all the data in the two data sets described in Section 2. The second ontology depicted in part B contains ten concepts, **Website Categories** and the subsets of the websites in each different category, namely, **Finance**, **Game**, **Sports**, **Computing/IT**, **Market**, **Knowledge**, **Telecommunication**, **Travel/holiday** and **News**. These are disjoints sets. The instances are those websites selected by the students in each category.

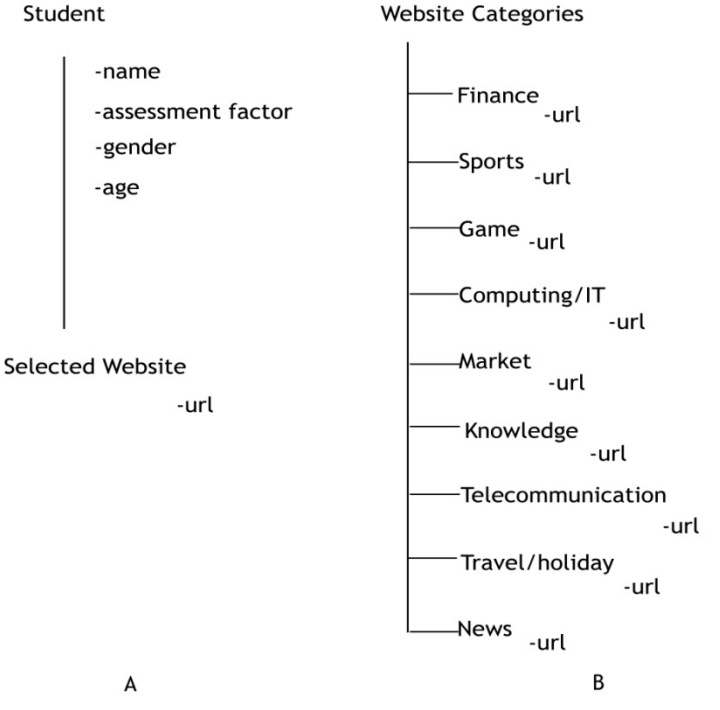


Fig. 1. Shows the outlines of the two ontologies. Unlike most of the cases, these two ontologies are for two completely different domains.

4 The Matching

Matching was made between a student and a website category. Unlike most of the cases, for example [5], these two concepts are completely different. Given two ontologies, the ontology-matching problem is to find semantic mappings between them. A mapping may be specified as a query that transforms instances in one ontology into instances in the other [5]. There is a one-to-one mapping between a website selected by a student and a website in one of those categories. The matching was used to build a relationship between a student and a website category. Similarity is used to measure the closeness of the relationship.

The attributes of a student was analyzed. Age was divided into three ranges: High (>24), Middle (in between 20-24) and Low (<20). Assessment Factor was divided

into three ranges as well: High (>1.05), Middle (in between 0.90-1.05) and Low (<0.90). A gender obviously includes two ranges: male and female. The second data set was used to learn the probabilities of falling each website category for these ranges. For a specific website category C and a range R, if total instances in C is TN, and RN of them are in range R, then the probability for instances in range R falling to category C is

$$P_R(C) = RN/CN \text{ where C is in } \{FI, GA, SP, CO, MA, KN, TE, TR, NE\} \quad (1)$$

Given an instance I, if its age range is RA, assessment factor range is RF, gender range is RG, and then the similarity measure between I and category C is

$$SM_C(I) = P_{RA}(C) + P_{RF}(C) + P_{RG}(C)$$

where C is in {FI, GA, SP, CO, MA, KN, TE, TR, NE} (2)

The larger $SM_C(I)$, the closer I and C, which means student I is more likely to choose a website in category C. The above formula (2) was tested by using the first data set. Three categories were used for the testing: Sports, Travel/Holiday and Game.

Table 4. The test results for Sports category

Samples	SM_{FI}	SM_{GA}	SM_{SP}	SM_{CO}	SM_{MA}	SM_{KN}	SM_{TE}	SM_{TR}	SM_{NE}
28M1.13	1.50	1.20	1.83	1.20	1.50	1.00	1.00	1.00	1.00
23M1.11	1.00	1.60	1.67	1.60	1.75	2.00	2.00	1.25	0.50
23M0.79	0.50	1.40	1.34	1.60	1.50	2.50	1.50	1.00	0.58

Tables 4, 5 and 6 show the test results. The Sports category has the best test results in general, the Travel/Holiday second, and the Game category. However, sample 2 in the Game category has the second best similarity in the whole row and sample 4 has the second worst similarity in the whole row. So the test results in this category are pretty diverged.

Table 5. The test results for Travel/Holiday category

Samples	SM_{FI}	SM_{GA}	SM_{SP}	SM_{CO}	SM_{MA}	SM_{KN}	SM_{TE}	SM_{TR}	SM_{NE}
23M1.05	1.00	1.60	1.50	1.40	2.25	2.50	2.00	1.75	0.00
38F1.18	1.50	0.60	0.83	1.00	0.50	0.00	1.00	1.00	1.50
48M1.10	1.50	1.20	1.83	1.20	1.50	1.00	1.00	1.00	1.00
35M0.98	1.50	1.20	1.66	1.00	2.00	1.50	1.00	1.50	0.50
21M1.15	1.00	1.60	1.67	1.60	1.75	2.00	2.00	1.25	0.50

Table 6. The test results for Game category

Samples	SM_{FI}	SM_{GA}	SM_{SP}	SM_{CO}	SM_{MA}	SM_{KN}	SM_{TE}	SM_{TR}	SM_{NE}
25M0.81	1.00	1.00	1.50	1.20	1.25	1.50	0.50	0.75	1.08
19M0.81	1.00	1.60	1.67	1.20	1.25	1.50	0.50	0.75	1.08
21M1.07	1.00	1.60	1.67	1.60	1.75	2.00	2.00	1.25	0.50
26F1.04	1.50	0.60	0.66	0.80	1.00	0.50	1.00	1.50	1.50

5 Discussion and Future Work

Although the test results are not accurate enough, some instances did get accurate results. The test results in Travel/Holiday category are reasonably close in general. So this pilot is promising. The small data size of this study makes it inappropriate to assess the accuracy at this stage. Improving the data sample size may bring more potential to this study. For large data size, an automatic process is necessary. The way how age range and assessment factor range were divided was based on experiment which requires further investigation. Further investigation is also required for the similarity formula.

This study only starts the investigation from a few basic personal attributes, which may be extended to cover more valuable and meaningful Internet related personal attributes. Collecting user interaction data on the selected websites may provide more information for us to understand the users and to help the users to understand more about the websites [6]. [4] suggests the similar ideas, such as providing personality testing tools on the website to build user profiles, tracing user behaviours allow better understanding on users' preference and interests.

The scope of this study is not limited to e-learning system. The similar approach could be applied to other online systems such as online shopping system, online banking system and etc. It might be particularly helpful to address the Internet security issues such as spam/malicious emails, spoofing, interception of sensitive data, data alteration, and denial of services, which are identified in most of the online systems. [7] reported that people under stress or other psychological pain were distinctively self-focused in their writing. This kind of information could be gathered by an automatic agent to identify potential sources of Internet security issues. [4] reported that risk-taking theory may help to explain which personalities are more likely to be willing to try new services that have a potential or an image of involving some degree of risk or insecurity. [8] reported a method that using data mining approaches for intrusion detection. In this method, two algorithms were used to automatically classify data with abnormal behaviors.

The future work of this study in Internet security area can be outlined as the following:

1. Collect user personal attributes data in two ways: online personality test and automatic agent.
2. Analyse the collected data to generate two ontologies: one for user personality; another for online security issues.
3. Analyse the collected data to find appropriate data mining algorithms to match the user personality ontology into the online security ontology.

References

1. Zouaq, A., Nkambou, R.: Building Domain Ontologies from Text for Educational Purposes. IEEE Transaction on Learning Technologies 1(1) (2008)
2. Fruhmann, K., Nussbaumer, A., Albert, D.: A Psycho-Pedagogical Framework for Self-Regulated Learning in a Responsive Open Learning Environment. In: Proceedings of the International Conference eLearning Baltics Science, Rostock,Germany, July 1-2 (2010)

3. Aroyo, L., Dicheva, D.: The New Challenges for E-Learning:The Educational Semantic Web. Educational Technology & Soc., 7(4), 59–69 (2004)
4. Amichai-Hamburger, Y.: Internet and personality. Computers in Human Behavior 18, 1–10 (2002)
5. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to map between ontologies on the semantic. Web. In: Proceedings of the International World Wide Web Conference, Honolulu, pp. 662–673 (2002)
6. Chi, E.H., Pirolli, P., Pitkow, J.: The scent of a site: A system for analyzing and predicting information scent, usage, and usability of a web site. In: Proceedings of the Conference on Human Factors in Computing Systems, pp. 161–168. The Hague, ACM Press, The Netherlands, New York (2000)
7. Barak, A., Miron, O.: Writing characteristics of suicidal people on the Internet: A. psychological investigation of emerging social environments. Suicide and Life-Threatening Behavior 35(5), 507–524 (2005)
8. Lee, W., Stolfo, S.: Data mining approaches for intrusion detection. In: Proceedings of the 7th USENIX Security Symposium, San Antonio, TX, pp. 59–69 (1998)

Entropy Based Discriminators for P2P Teletraffic Characterization

Tao Ban, Shanqing Guo, Masashi Eto, Daisuke Inoue, and Koji Nakao*

National Institute of Information and Communications Technology
4-2-1 Nukui-Kitamachi, Tokyo, 184-8795, Japan
bantao@nict.go.jp
http://www.nict.go.jp

Abstract. Characterization of P2P traffic is an essential step to develop workload models towards capacity planning and cyberthreat countermeasure over P2P networks. In this paper, we present a new scheme for monitoring and characterizing File-Sharing P2P (FSP2P) applications. Featured by both lightweightness and high prediction accuracy, the proposed scheme supports performance tuning between monitoring cost and the system response time, and is adaptable to network environments with different specifications.

Keywords: Application behavior analysis, network monitoring, P2P protocol.

1 Introduction

Recent studies on P2P traffics [1] show that (1) the proportion of P2P traffics is now prevalent over the Internet and keeps increasing; (2) many P2P applications are bandwidth-intensive, resulting in excessive network congestion and possible dissatisfied customers and customer churn; (3) P2P file-sharing always causes controversy because of legal issues such as copyright law violations; and (4) most P2P clients are vulnerable to attacks and when compromised will lead to serious information leak and other catastrophic problems.

Appropriate capacity planning and malware countermeasure could only be available when P2P traffic is identified with high accuracy. However, traditional monitoring and analysis techniques do not support analysis of evolving P2P network applications that show more sophisticated characteristics. Towards a tractable solution to mitigation of the bandwidth intensity of P2P protocols and proactive detection of malware proliferation over P2P networks, in this study, we present a cost-effective scheme to characterize File Sharing P2P (FSP2P) applications. To enable the lightweightness and adaptability of the proposed scheme, instead of costive inspection into the packet payload, we make use of meta features extracted from the header fields of the packets. To grasp the overall one-to-many topological nature of a P2P communication, analysis is done at

* This research is partially supported by the Key Science-Technology Project of Shandong Province, P. R. China (Grant No. 2010GGX10117).

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 18–27, 2011.
© Springer-Verlag Berlin Heidelberg 2011

the host level, which is different from previous approaches that focus on flow level. Moreover, utilization of entropy based features that well represents the distributedness of the P2P communication could further improve the prediction accuracy of the system. Experiments show that the proposed scheme, with efficient parameter tuning, could achieve satisfactory prediction accuracy with manageable costs on network bandwidth and computing resources.

The rest of this paper is organized as follows. Section 2 presents previous work on P2P teletraffic monitoring and characterization. Section 3 introduces the formulation of a classification problem for P2P file sharing host analysis. Section 4 reports experimental results based on the proposed scheme. Section 5 draws the conclusion.

2 Related Work on P2P Network Characterization

Network traffic monitoring and analysis has been a handy tool to fight against cyberthreats and prevent abusive or illegal resource usage. Depending on which network layer they operate at, P2P traffic monitoring systems could be largely grouped into network-level tracing and application level tracing.

2.1 Network Level Tracing

Network-level tracing usually refers to IP-level packet monitoring at suitable points in the network infrastructure. Because of its transparency to the P2P network and capability to characterize multiple P2P systems simultaneously, network-level tracing has been the main research stream. Previous work on transport layer [2] suggests that transport layer information could still manage to identify a considerable part of P2P traffic. Another approach to characterize P2P transmission sessions is packet-payload examination [3]. Despite its powerfulness, payload inspection usually encounters legal, privacy, and financial obstacles, as well as subjects to some technical problems of costive reverse engineering and payload decryption. Other available approaches to P2P protocol characterization include signature-based methods [4], which use a portion of payload information to extract the distinguishable information of different protocols; flow-level analysis engines [5], which are based on the statistics or patterns of the packet flows; and hybrid methods [6], which use heuristics such as well known port numbers, frequently observed signatures in 16-byte payload, and source and destination IP addresses for the characterization.

2.2 Application Level Tracing

Depending on the characteristics to be discovered, one can also apply application-level tracing tools to trace and analyze the traffic of a specific application. According to the mode the monitoring program works in, application-level tracing methods are classified into passive application-level tracing – tracing is performed by monitoring the resource discovery and network maintenance messages

that a P2P host sends and receives while it communicates with other peers at the application level, and active application-level tracing which employs an aggressive querying and connection policy so that the monitoring peer attempts to connect to and interrogate as much of the P2P network as possible.

2.3 Hybrid Virtualization Based Approach

The previous work in [7], strikes a balance between all the approaches mentioned above. To offer insights into a specific P2P network, a one-protocol-exclusive network is configured in [7] for network-trace collection. Advantages of such an exclusive network are summarized in the following aspects. First, application specific characteristics can be abstracted from the collected traces. Second, collected traces are automatically labelled with good reliability and little labor cost, ready for further supervised analysis. Moreover, because the traces are collected at the network-level, the system built for one P2P network can be easily reusable for any other (P2P) protocol.

The system introduced in [7] are composed of three layers. The network layer offers accessibility to the outside network and high-performance storage service for the upper layers. The server layer offers a virtualization environment to the guest OSes, captures the individual traces for each of them, and send the traces to the storage server on the network layer. The virtual machine layer is characterized by guest OSes where specific P2P clients are installed with Internet connection enabled. Via the Virtual Machine (VM) technology, this hybrid approach is featured by the following advantages. First, multiple OSes could coexists on the same computer, which enable us to build a comparably large-scale network environment without the burden to handle with so many physical machines. Second, it makes more efficient usage of the computer resources. With the available resources at hand, we can achieve a much larger network environment. Then, in case the guest OSes are compromised by malware, VM can help to sandbox the OSes, rendering the hypervisor safe from harm. The last but not least important, thanks to the fast system recovery and reboot capability of the VM technology, it is much easier to redo the experiment or adapt the system to analyze other P2P protocols.

3 Data Analysis

In this section, we describe the method to analysis the network traces collected by the proposed system. The objective is to differentiate the hosts which are doing P2P file sharing from those who are running other network applications. We formulate the problem as a supervised learning problem, i.e., traces are divided into two classes with the positive class standing for P2P file sharing hosts and the negative class being other hosts. Hereafter, we refer a host which is doing P2P file sharing as a P2P node, where the term P2P is used in a narrow sense.

3.1 Host-Level Analysis

Our analysis makes use of statistical meta features, referred as discriminators hereafter, extracted from the monitored interfaces and detects anomalies associated with P2P nodes. By only using packet header information alone, we mitigate the collection-computation costs of an alternative signature based approach, and avoid privacy related issues which could arise from traffic-payload inspection.

Traffic anomalies on metrics such as the traffic volume, packet size, number of preserved connections are often good indicators of P2P applications. Previously, these discriminators are often derived on *network flows* – traffic channels between communicating peers defined by the 5-tuple, i.e., source address, destination address, source port, destination port, protocol. The definition of traffic flow well suits traditional network functions such as email exchange, web access, and database access, which are built on the Client-Sever (CS) model, and thus is effective in analyzing the Internet's mainstream application protocols, such as HTTP, SMTP, Telnet, and DNS. However, recent P2P applications tend to hide their appearances by disguising as ordinary network transmissions such as web browsing or FTP file transferring. Therefore, at the network flow level, there could be little difference between a P2P flow and a web browsing flow or an FTP flow, because all such applications tends to maximize the bandwidth usage for better user experience. Therefore, flow level statistics as well as information such as port numbers and protocol types cannot offer sufficient information for discriminating a P2P flow from flows associated with ordinary CS type applications. Consequently, loss of discriminating information in these discriminators will result in inefficacy of analysis engines.

In this study we are interested in the status of a host, i.e., whether it is doing P2P file sharing, rather than the characteristics of its specific communication session/channel. For better capturing the decentralized nature of a P2P network, in previous work [8], we proposed to go beyond the flow-level to the host-level, where an overall picture of the communication is readily available. To do so, we treat all the communications bounded to a target host (associated with a specific IP address) as a single stream and define discriminators upon these host-level streams.

3.2 Discriminators

It is well-known that, the performance of a learning system largely depends on the discriminant ability in the discriminators extracted from the underlying process. For more accurate P2P node characterization, it is expected that a discriminator vector extracted from a P2P node will show statistical difference from vectors associated with ordinary nodes, while discriminators for different P2P applications will share common characteristics so that they are logically close in the feature space.

In our previous study [8], we have made use of discriminators that are extracted on basis of data traces collected on a node during a fixed time window: number of packets, payload volume, time range, payload speed, number of protocols, number of source IPs, number of destination IPs, number of source ports,

number of destination ports, and number of TCP flags. In addition to the above classical discriminators which well represent the bandwidth-intensive nature of P2P streams, here we pay special attention to the topological characteristics of P2P file-sharing protocols. To treat with instability and variable connectivity, all P2P file sharing application tend to keep a large amount of connections between peering nodes. Thus the distributedness of the transmission, i.e., the one-to-many transmission mode from a host upon the peering nodes will be a key feature differentiating a P2P stream from ordinary CS-mode streams.

An ideal discriminator that describes the distributed characteristic of a flow is the *entropy*. In information theory, the *entropy*, or *Shannon entropy*, is a measure of the uncertainty associated with a random variable. In our case, it measures the randomness of given attributes in the network packets' headers. These attributes could be the source IP address, the source port, protocol type or any other properties of interest that could contribute to differentiating the two classes. Take the source IP addresses appeared in a fixed time window as an example. The entropy, H, over the IP-address space is calculated is as [9]

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i, \tag{1}$$

where n is the number of distinct IP addresses and p_i is the probability that the ith IP address shows up as source IP during that period.

Fig. 1 shows the scatter plots in the multi-dimensional space defined by all discriminators we have defined. It is easy to spot the separability of the two classes within in the subspace defined by randomly paired dimensions: data from the positive class (shown as green markers) and data from negative class (shown as blue markers) forms perceptible classification boundaries.

3.3 Learning Methodology

Despite that many analysis models, e.g., clustering, function regression, association rule mining, etc., could help to characterize the host behavior based on the collected network traces, the classification model best fits the objective of this study – to identify whether a host is performing P2P file sharing operations at the moment. As our objective is to separate P2P nodes from ordinary nodes, the task turns out to be a *binary classification* problem. That is, we are given empirical data

$$(x_1, y_1), ..., (x_m, y_m) \in \mathcal{X} \times \{\pm 1\}, \tag{2}$$

where $\mathcal{X}$ is the nonempty set of all possible *observations* x_i (in our case, x_i are numerical discriminator vectors extracted from the network stream within a predefined time window); $y_i \in \{\pm 1\}$ are called class labels, where $+1$ stands for a P2P node and -1 stands for a non-P2P node. By using a classification algorithm, we are going to build a *decision function*, $f(x)$, so that when applied to a discriminator vector associated with a new coming host, it will predict whether it belongs to class $+1$ or class -1.

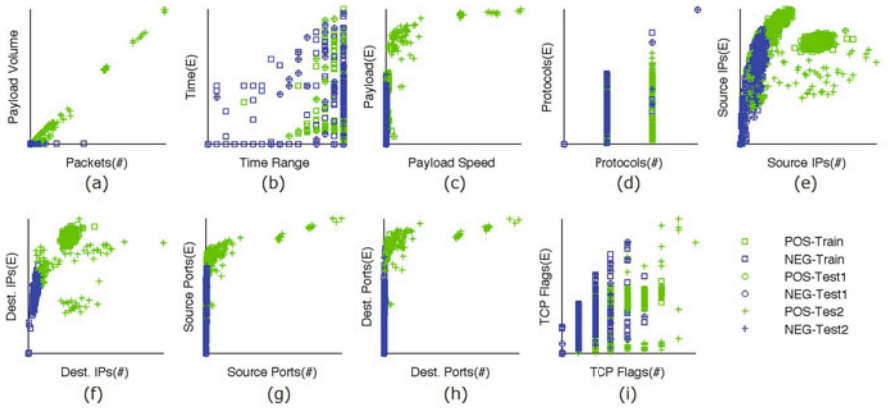


Fig. 1. Scatter plots in the multi-dimensional space defined by discriminators. (#) stands for *number* and (E) stands for *entropy*.

In the experiments, we adopted the Support Vector Machine (SVM) [10] for training and prediction. Intuitively, an SVM model is built by finding the linear separating hyperplane which realizes the maximal margin criterion, i.e., the observations from different classes are divided by a clear gap that is as wide as possible. The class labels of new observations are predicted based on which side of the separating hyperplane they fall on. When it happens that in the input space the sets to be discriminated are not linearly separable, the original dimensional space could be mapped into a space with much higher dimension to make the separation easier in that space. Using the so called *kernel trick*, this mapping could be done in an explicit way so that the computational load is kept to a reasonable extent. SVM has enjoyed high popularity in many application areas because of its generality and high prediction performance. More details about SVM can be found in [10] (not included because of page limit).

4 Experiments

To testify the feasibility and evaluate the performance of the proposed scheme, we apply it to traffic analysis in the following settings.

4.1 Experiment Settings

In the first setting, the system performs classification between the background traffic and two most popular P2P protocols, i.e., BitTorrent, which is the world's most popular P2P file sharing protocol, and PPLive, which is a typical protocol of the new generation P2P applications known as P2PTV. Training and test are both performed on trace data containing the same protocols. Hereafter we refer this first test set as test set 1 (TS-1 in figures).

In the second setting, in addition to accessing the classification performance of known P2P protocols, the system's generalization ability to previously unknown protocols is also taken into consideration. In order to do so, training is performed using the same trace data as that is used in the first setting, however, in the test data, we add traces generated from previously unknown P2P protocols. Namely, we add traces of eMule – a popular protocol for P2P file sharing, and PPStream – yet another P2PTV protocol, to the test data. Hereafter we refer this second test set as test set 2 (TS-2 in figures).

The P2P traces are collected within the following network environment. For each of the eight-core servers, we run 6 virtual machines with Windows 2000 installed. QEMU is chosen as the hypervisor software for performance reason. Related software clients are installed and run upon the guest OSes. A P2P trace is collected and labeled based on what client is running on the guest OS. The background traffic is captured in a similar network environment where common network operations such as web browsing, FTP/HTTP downloading, and online gaming are permitted.

In SVM training, we use the Gaussian kernel which is a universally adaptable kernel function with reliable performance in a wide range of applications. Parameter settings, i.e., width parameter of the kernel, γ, and error penalty parameter, C, are determined by 10-fold cross validation. All the reported results are averaged on 100 runs on the randomly shuffled versions of the same trace set.

4.2 Analysis on Time Window Size

In the first experiment, we explore the influence of the time-window size, w, on the prediction accuracy. To justify whether a host is a P2P node, traffics bounded to this host needs to be monitored for at least a time period of w (seconds) so that discriminators could be extracted on the captured trace over this period. In this sense, the window size is closely related to the response performance of the system. To access its influence, w is selected from $\{1, 2, 4, 8, 16, 32, 64\}$ (seconds), and as w changes its value, the variation of prediction accuracy on the two test sets are recorded and shown in Fig. 2(a) and 2(b), respectively.

In Fig. 2(a) the prediction accuracy (vertical axis) on test set 1 is plotted against w (horizontal axis). It is easy to spot that the discriminant information in the discriminators gradually increases as w increases from $1s$ to $64s$. When the sampling rate is 1, i.e., all captured data are used for feature extraction, the prediction rate increases from 95.57% ($w = 1s$) to 99.67% ($w = 64s$). Lowering the sampling rate, which is noted as a r parameter hereafter, leads to a degeneration of the accuracy to some extent. Still, in all cases, the accuracy increases as the window size grows. When $w = 64s$, the accuracy approaches 99.48% for a sampling rate of $\frac{1}{8}$ and 97.29% for a sampling rate at $\frac{1}{64}$, respectively.

The results on test set 2 is shown in Fig. 2(b). Generally, including previously unknown protocols in the test set will render the data distribution different from that of the training data and thus results in a degeneration of the prediction accuracy. The results in Fig. 2(b) shows that the conclusion that increment of

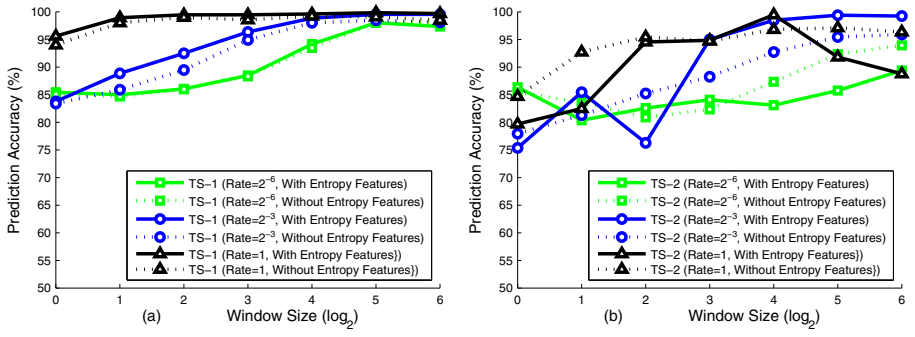


Fig. 2. Prediction accuracy v.s. window size

w leads to higher prediction accuracy also applies to previously unknown P2P protocols, despite of more obvious fluctuations. When $w = 64s$, all the settings with different sampling rates achieve an accuracy rate above 88.75%. For $r = \frac{1}{64}$, the increment of the accuracy appears to be very stable, in spite of a drop at the starting point. For $r = \frac{1}{8}$, the rate optimal accuracy is 99.37% at $w = 32s$. For $r = 1$, it goes to a maximum of 99.39% at $w = 16s$, and decrease to 88.76% because of the deficiency of the classifier.

As a reference, the dotted lines in Fig. 2(a) and 2(b) show the prediction accuracy without using the entropy discriminators. In Fig. 2(a), it is easy to spot that for test set 1, including the entropy discriminators could always increase the separability of the two classes, resulting in increased prediction accuracies. In Fig. 2(b), entropy discriminators work very well in some of the cases, e.g., when $r = \frac{1}{8}$; in some other cases, they lead to some degeneration of the prediction accuracy because of increased complexity they have introduced.

4.3 Analysis on Sampling Rate

When monitoring high speed switched networks, the sampling rate r on the network interface is one of the most important parameters that determine the scalability of the monitoring. Generally, we want to reduce the sampling rate for cost effective traffic data collection, storage, and analysis. The second group of experiments are designed to verify the influence of sampling rate on the prediction accuracy. To do so, the traces are first captured using full sampling, i.e., $r = 1$, followed by a subsampling procedure using different r parameters selected from $\{\frac{1}{64}, \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1\}$. Then numerical discriminators are extracted from the subsampled traces and forwarded to the classifier for training. As r changes its value, the variation of classification accuracy on the two test sets are recorded and shown in Fig. 3(a) and 3(b), respectively.

Fig. 3(a) plots the prediction accuracy against sampling rate on test set 1. The figure looks similar to Fig. 2(a). At this time, the increment in sampling rate contributes to the increment in prediction accuracy. For $w = 1s$, the accuracy starts from 85.41% at $r = \frac{1}{64}$, slightly dropping to a minimum of 83.78% at

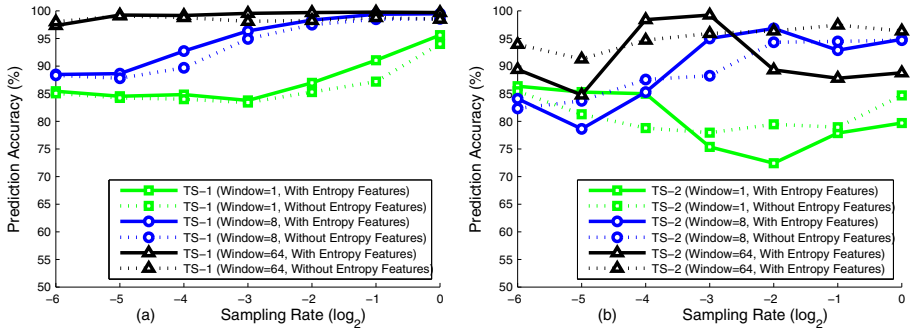


Fig. 3. Prediction accuracy v.s. sampling rate

$r = \frac{1}{8}$, and then increases gradually to 95.57% at $r = 1$. For $w = 8s$, the accuracy constantly increases from 88.44% at $r = \frac{1}{64}$ to 99.41% at $r = 1$. For $w = 64s$, except the beginning point at $r = \frac{1}{64}$, all other r values all give accuracy above 99.14%.

Fig. 3(b) shows the results on test set 2. We can easily observe the analogy between this figure and Fig. 2(b). Discordance between the training set and test set has led to larger variations in the prediction accuracy. Still, we can find a trend that increment in the sampling rate generally leads to raises in prediction accuracy, except for the case of $w = 1s$.

As can be seen in Fig. 3(a), including entropy discriminators could stably improve the prediction accuracy on test set 1. For test set 2, the best accuracy is obtained with entropy discriminators at $(w = 64s, r = \frac{1}{8})$, nevertheless, entropy discriminators also lead to remarkable fluctuations in prediction accuracy as other parameters change.

To summarize, the above experiments on windows size and sampling rate show that, more computing resources, i.e., longer observation time and/or higher sampling rate will provide us more knowledge about the behavior of the monitored host, and thus results in higher prediction accuracy – as typically experienced in all computer science related fields. Another important discovery is that, the positive effect of increasing the window size of observation excesses the negative effect of decrement in sampling rate. This suggests that for better recognition accuracy and less system performance downgrade, we can keep a rather small sampling rate for network performance purpose while increase the window size until satisfactory generalization performance is achieved. The entropy discriminators adopted in the study is remarkable in the following two aspects. First, for prediction on known protocols (test set 1), stable gain in prediction accuracy is guaranteed by incorporating entropy discriminators. Second, when used for predicting previously unknown protocols (test set 2), the optimal result are always obtained using these discriminators although they might also introduce some fluctuations when other parameters change. Above all, when the entropy discriminators are also taken into consideration, $w = 64s$ and $r = \frac{1}{16}$ seems to be a very good parameter combination that supports good accuracy on prediction

of P2P hosts (99.14% on test set 1 and 98.38% test set 2), without imposing a heavy impact on the monitored network.

5 Conclusion

In this paper, we have presented a study on applying machine learning techniques for characterizing P2P file sharing hosts in the network for network engineering purpose. For better lightweightness and adaptability, we define informative discriminators based on the headers of the packets instead of deep payload inspection. For better accuracy, we propose to perform the analysis at host level so that entropy based discriminators that capture the one-to-many topological nature of a P2P file-sharing transmission could be defined. Numerical study showed that with appropriate parameter tuning the proposed scheme could realize lightweight (with $\frac{1}{16}$ sampling rate) and accurate prediction with accuracy above 98% even for previously unknown P2P protocols.

References

1. http://www.ipoque.com/news_&_events/internet_studies/internet_study_2007
2. Karagiannis, T., Broido, A., Faloutsos, M., Klaffy, K.: Transport Layer Identification of P2P Traffic. In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement, pp. 121–134 (2004)
3. Kang, H., Ju, H., Kim, M., Hong, J.W.: Towards Streaming Media Traffic Monitoring and Analysis. In: Proceedings of 2002 Asia-Pacific Network Operations and Management Symposium, pp. 97–108 (2002)
4. Haffner, P., Sen, S., Spatscheck, O., Wang, D.: ACAS: Automated Construction of Application Signatures. In: Proceedings of the ACM SIGCOMM Workshop on Mining Network Data, pp. 107–202 (2005)
5. Sen, S., Wang, J.: Analyzing Peer-to-Peer Traffic Across Large Networks. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, pp. 137–150 (2002)
6. Bolla, R., Rapuzzi, R., Sciuto, M.: Monitoring and Classification of Teletraffic in P2P Environment. In: Proceedings of the 2006 Australian Telecommunication Networks and Application Conference, pp. 313–318 (2006)
7. Ban, T., Ando, R., Kadobayashi, Y.: Monitoring and analysis of network traffic in P2P environment. NICT Journal 54(2/3), 31–39 (2008)
8. Ban, T., Guo, S., Zhang, Z., Ando, R., Kadobayashi, Y.: Practical Network Traffic Analysis in P2P Environment. In: 2nd International Workshop on TRaffic Analysis and Classification(TRAC 2011), July 5–8 (2011)
9. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. In: ACM SIGCOMM (August 2005)
10. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)

Faster Log Analysis and Integration of Security Incidents Using Knuth-Bendix Completion

Ruo Ando and Shinsuke Miwa

National Institute of Information and Communication Technology,
4-2-1 Nukui-Kitamachi, Koganei,
Tokyo 184-8795 Japan
ruo@nict.go.jps
http://www.nict.go.jp/

Abstract. With the rapid popularization of cloud computing, mobile devices and high speed Internet, recent security incidents have become more complicated which imposes a great burden on network administrators. In this paper we propose an integration and simplification method of log strings obtained by many kinds of computer devices: memory, socket and file. Besides, we apply reasoning strategy for term rewriting called as Knuth-Bendix completion algorithm for ensuring termination and confluent. Knuth Bendix completion includes some inference rules such as lrpo (the lexicographic recursive path ordering) and dynamic demodulation. As a result, we can achieve the reduction of generated clauses which result in faster integration of log strings. In experiment, we present the effectiveness of proposed method by showing the result of exploitation of vulnerability and malware's behavior log.

Keywords: Log analysis and integration, mechanized reasoning, Knuth Bendix completion, lrpo, dynamic demodulation.

1 Introduction

With the rapid popularization of cloud computing and high speed Internet, recent security incidents have become more complicated which imposes a great burden on network administrators. Once the security incidents has been occurred, network administrators should check event log string which is composed of many kinds of devices and software layers. Log integration is important to retrieve information from large and diversified log strings. We cannot find evidence and root cause of security incidents from one kind of event logs. For example, even if we can obtain fine grained memory dumps, we cannot see which direction the malware infecting our system sends packets.

In this paper we propose a faster method to generate log rewriting system for translating a set of log strings A into B. In this case, a set of log strings B should be informative for users and there should be an evidence inside B. For this purpose, we apply Knuth Bendix completion algorithm for fastening log integration process. Knuth bendix completion algorithm provides two properties with reasoning system: confluent and ensuring termination. In experiment, we show the

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 28–36, 2011.
© Springer-Verlag Berlin Heidelberg 2011

effectiveness of Knuth Bendix completion for faster log integration. This paper is composed as follows. In chapter 2, we have outlined what is mechanized reasoning. Some inference rules are introduced. Knuth Bendix completion algorithm is illustrated briefly in chapter 3. In chapter 4 and 5, we present experimental results concerning vulnerability exploitation and malware log analysis. We discuss the effectiveness of applying Knuth Bendix completion in chapter 6 and then conclude with this paper in chapter 7.

2 Mechanized Reasoning

Mechanized reasoning is also called as automated reasoning in which fields researchers cope with the creation of software which makes computers "reason" in the sense of mathematical aspects such as solving puzzle and proving theorems. In this field, software such as FoL (first-order logic) or HoL (higher order logic) theorem prover, SAT solver and model generator is created to automated the mathematical process. Also, weighting clauses for resolution [1] and effective term rewriting techniques [2] are applied for mechanized reasoning. In this paper we leverage automated reasoning for log analysis and integration.

2.1 Resolution

Basically, proposed system is based on resolution. If clause Cls1 and Cls2 have literal L1 and L2, the clause C_R is resolved below.

$$C_R = (C_1\sigma \setminus L_1\sigma) \cup (C_2\sigma \setminus \bar{L_2}\sigma)$$

where σ is unifier which L1 and L2 equal. σ is sometimes could be most general unifier. The resolution in $Lit_1 \in Cls_1$ is also possible as $Lit_n \in Cls_n$. According to the above formation, hyper resolution several clauses.

2.2 Set of Support Strategy

Set of support was introduced by L.Wos, S.Robinson and Carson in 1965 [3]. If the clause T is retrieved from S, SOS is possible with the satisfiability of S-T. Set of support strategy enable the researcher to select one clause characterizing the searching to be placed in the initializing list called SOS. For the searching to be feasible and more effective, the resolution of more than one clauses not in SOS is inhibited in order to prevent the prover go into abundant searching place. Definition. H is satisfiable subset of S. T is set of support where

$$S = \cup H \text{ and } H \cap T = \phi$$

Figure1 show the resolution process in set of support strategy, where S=P and Q and R, P and R, Q and R, R. The restriction imposes the reasoning so that the program do not apply an inference rule to a set of clauses that are not the complement of set of support.

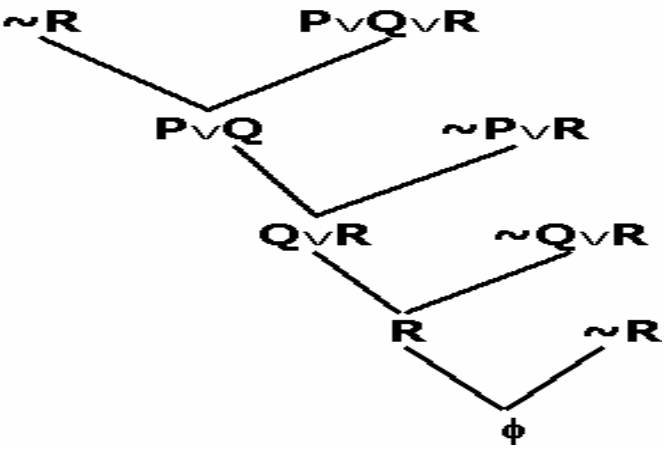


Fig. 1. Figure1 show the resolution process in set of support strategy, where S={P and Q and R, P and R, Q and R, R}.The restriction imposes the reasoning so that the program do not apply an inference rule to a set of clauses that are not the complement of set of support.

2.3 Hyperresolution

For effective resolution, we apply the inference rule called hyper resolution [4], which is a kind of resolution that can do resolutions at once compared with several steps in another rules. For hyperresolution, these must be the negative or mixed clause with the remaining clauses equal to the number of literals in the negative or mixed clause. The positive clause are described as satellites, the negative clause nucleus. "Hyper" means that in this resolution more process has occur than another resolution such as binary resolution.

2.4 Subsumption

Subsumption [5] is the process of discarding a specific statement. The clause that duplicated or is less general is discarded in the already-existing information. As a result, subsumption prevents a reasoning program from retaining clauses that is obviously redundant, especially is logically captured by more general clauses. Definition. The clause A subsumes the clause B when B is the instance that is logically captured by B.

```
The clause P(X) Subsumes the clause P(a).
```

There are several paths and axioms that could be applied. Subsumption strategy is effective when the same or more specific clause in the present of already-existing clause is generated. The clause is crossed and the generated clauses on the process of resolution is also eliminated. The effectiveness of this strategy is presented in experimental results.

3 Knuth-Bendix Completion

The Knuth-Bendix completion algorithm[6] is an algorithm for transforming a set of equations into term-rewriting system. The term rewriting system generated by Knuth-Bendix completion has two properties: confluent and ensuring termination.

Kunuth-Bendix Completion 1. *If a rewrite system S is Knuth-Bendix complete and terms* $\theta 1$, $\theta 2$ *and* η *in respect to S, S(*η*) is a result of rewriting of S(*$\theta 1$*) and S(*$\theta 2$*). Confluent:* $R \vdash$ *S(*$\theta 1$*) and S(*$\theta 2$*).*

In term rewriting system, equational reasoning which is also called as reduction does not guarantee that reasoning process will stop. There are two reasons for this. First, several inference rewriting rules can be applied for the same set of equations. Second, there could be unlimited sequence of reduction for rewriting. Confluent is the property for assuring that several inference rules go togher in the reasoning process. Ensuring termination means that the system does not go into the way of unlimited reduction process.

One example concerning Knuth-Bendix completion could be associative law.

$$(x\phi y)\phi z = x\phi(y\phi z)$$

Once Knuth-Bendix completion algorithm is applied, lrpo (the lexicographic recursive path ordering) inference rule is set. Also, demodulation could be generated. According to lrpo,

$$(x\phi y)\phi z -> x\phi(y\phi z)$$

Right side could be translated according to demodulation,

$$x\phi(y\phi z) -> (x\phi y)\phi z$$

As is now clear, Knuth-Bendix completion is compound strategy such as lrpo and dynamic demodulation.

4 Experiment I

For validation of the improvement of proposed system using Knuth-Bendix completion, we show two kinds of experimental results: detection and log integration. First, we represent the attack detection in strings logged by two kinds of exploitation: internet explorer (MS979352) and ftp server attack (CVE-1999-0256).

4.1 Internet Explorer Aurora Attack: MS979352

In this paper we cope with an exploitation of the vulnerability of Internet explorer which is called Aurora attack [7]. Aurora attack, described in Microsoft

Security Advisory (979352), is implemented for the vulnerability in Internet explorer which could allow remote code execution. Reproduction of aurora attack is done by Java script with attack vector on server side and Internet explorer connecting port 8080, resulting in the shell code operation with port 4444. In detail, authors are encouraged to check the article such as [8].

4.2 FTP Server Attack

FTP server attack we cope with in this paper is exploitation of buffer overflow included in warFTPD with CVE-1999-0256. This exploitation of WarFTPD is caused by the buffer overflow vulnerability allowing remote execution of arbitrary commands. Once the malicious host send payloads for warFTPD and exploitation is succeeded, attacker can browse / delete / copy arbitrary files in the remote computer. In detail, authors are encouraged to check the site such as [9].

Table 1. Attack log detction in IE and FTPD exploitation

inference rule	aurora attack	ftp server exploit
hyper resolution	6293	44716
hyper resolution with KBC	723 ∇	1497 ∇
binary resolution	6760	71963
binary resolution with KBC	1628 ∇	2372 ∇
UR resolution	6281	71963
UR resolution with KBC	723 ∇	4095 ∇

4.3 Results

In experiment, we use three kinds of inference rules: hyper resolution, binary resolution and UR (Unit Resulting) resolution. Table 1 shows the result of applying Knuth-Bendix Completion for six cases. In all cases, a number of generated clauses are reduced. In two cases, hyper resolution has best performance with 6293/723 and 44716/1497. Compared with aurora attack, ftp server attack has more impact of Knuth Bendix Completion Algorithm with reduce rate 97%, 97% and 95%.

5 Experiment II

In experiment II, we apply our method for integrating event log generated by Malware and extract information of fine-grained traffic analysis. We use MARS dataset proposed by Miwa. MARS dataset is generated and composed in starBED which is large scale emulation environment presented in [10]. MARS partly utilize Volatility framework to analyze memory dump for retrieving information of process. Also, we track packet dump using tcpdump for malware log.

5.1 Malware Log Analysis

Malware log is composed of two items: packet dump of each node and process behavior in each node.

```
Log format:
1: socket(pid(4),port(0),proto(47)).
2: packet(src(sip1(0),sip2(0),sip3(0),sip4(0),sport(bootpc)),
    dst(dip1(255),dip2(255),dip3(255),sip4(255),dport(bootps))).
3: library(pid(1728),dll(_windows_0_system32_comctl32_dll))
4:  file(pid(1616),file(_documentsandsettings_dic)).

Notes:
1: Socket information: which process X opens port number Y
   with protocol Z.
2: Packet information: what kind of packet is sent to
   this host X with port Y from address Z.
3: Library information: which process X loads library Y.
4: File information; which process X reads file X.
```

It is important to point out that the integration of [1] and [2] retrieves information about traffic log with process ID. Usually, packet information is dumped based on host-based, which result in the requirement of fine-grained process based traffic log. Resulting [1] and [2] with pid enables which process X generates traffic from Y to Z.

5.2 Results of Integration

In this section we present the experimental result of integrating log string generated by malware. We have tested proposed algorithm with Knuth-Bendix completion for 9 malwares. Table 2 shows the number of generated clauses in integrating malware logs. From malware #1 to #9, the average reduction rate

Table 2. The number of generated clause in integrating malware logs

malware ID	hyper res	hyper res with KBC	demod inf	demod inf with KBC
09ee	40729	16546 ∇	19180	12269 ∇
0ef2	39478	15947 ∇	14560	11754 ∇
102f	35076	11966 ∇	16356	10376 ∇
1c16	40114	15463 ∇	18950	4172 ∇
2aa1	40116	15944 ∇	19025	4194 ∇
38e3	40083	16062 ∇	14594	4237 ∇
58e5	39260	16059 ∇	14847	4217 ∇
79cq	40759	15662 ∇	14633	4061 ∇
d679	35290	16650 ∇	14462	1577 ∇

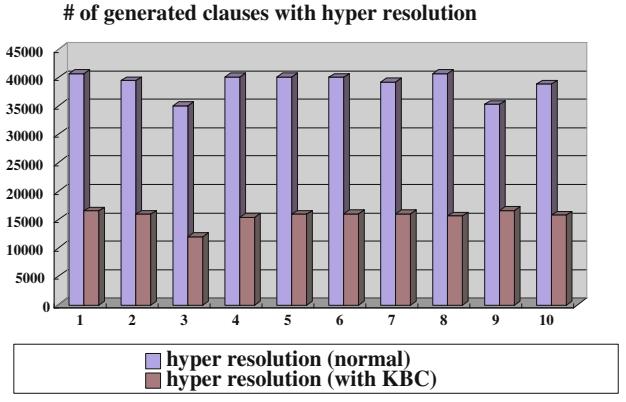


Fig. 2. The number of clauses generated in hyper resolution. In avarage, Knuth- Bendix completion algorithm reduces the automated reasoning cost about by 60%.

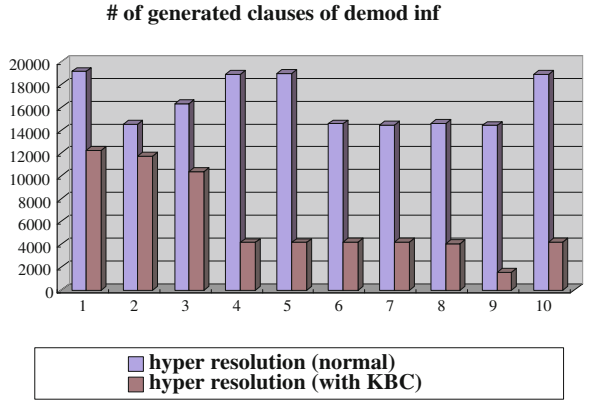


Fig. 3. The number of clauses generated in demodulation. Compared with hyper resolution, the effectiveness of Knuth-Bendix completion is heterogeneous from about 40% (#1) to 90% (#9).

is about 60%. Figure 2 and 3 dipicts the number of generated clauses in hyper resolution and demodulation. The number of clauses generated in demodulation. Compared with hyper resolution, the effectiveness of Knuth-Bendix completion is heterogeneous from about 40% (#1) to 90% (#9).

6 Discussion

In experiment, we have better performance when the Knuth-Bendix completion algorithm is applied for our term rewriting system. In the case of detecting attack logs of vulnerability exploitation, the effectiveness of Knuth-Bendix completion

changes according to the attacks and inference rules. Also, the reduction rate of generated clauses are various to nine kinds of malwares, particularly concerning demodulation which play a important role. Proposed method has been improved from our previous system in [11]. In general, we did not find the case where Knuth-Bendix completion increases the number of generated clauses. Ensuring termination and confluent are fundamental and key factor when we construct automated deduction system. It is not the case when we apply inference engine for integrating malware logs.

7 Conclusion

In this paper we have proposed the method for integrating many kinds of log strings such as process, memory, file and packet dump. With the rapid popularization of cloud computing, mobile devices high speed Internet, recent security incidents have become more complicated which imposes a great burden on network administrators. We should obtain large scale of log and analyze many files to detect security incident. Log integration is important to retrieve information from large and diversified log strings. We cannot find evidence and root cause of security incidents from one kind of event logs. For example, even if we can obtain fine grained memory dumps, we cannot see which direction the malware infecting our system sends packets. In this paper we have presented the method for integrating (and simplifying) method of log strings obtained by many devices. In constructing term rewriting system, ensuring termination and confluency is important to make reasoning process safe and faster. For this purpose, we have applied reasoning strategy for term rewriting called as Knuth -Bendix completion algorithm for ensuring termination and confluent. Knuth bendix completion includes some inference rules such as lrpo (the lexicographic recursive path ordering) and dynamic demodulation. As a result, we can achieve the reduction of generated clauses which result in faster integration of log strings. In experiment, we present the effectiveness of proposed method by showing the result of exploitation of vulnerability and malware's behavior log. In experiment, we have achieved the reduction rate about 95% for detecting attack logs of vulnerability exploitation. Also, we have reduced the number of generated clauses by about 60% in the case of resolution and about 40% for demodulation. For further work, we are planning to generalize our method for system logs proposed in [12].

References

1. Wos, L.: The Problem of Self-Analytically Choosing the Weights. J. Autom. Reasoning 4(4), 463–464 (1988)
2. Wos, L., Robinson, G.A., Carson, D.F., Shalla, L.: The Concept of Demodulation in Theorem Proving. Journal of Automated Reasoning (1967)
3. Wos, L., Robinson, G.A., Carsonmh, D.F.: Efficiency Completeness of the Set of Support Strategy in Theorem Provingh. Journal of Automated Reasoning (1965)
4. Wos, L.: The Problem of Explaining the Disparate Performance of Hyperresolution and Paramodulation. J. Autom. Reasoning 4(2), 215–217 (1988)

5. Wos, L.: The Problem of Choosing the Type of Subsumption to Use. J. Autom. Reasoning 7(3), 435–438 (1991)
6. Knuth, D., Bendix, P.: Simple word problems in universal algebras. In: Leech, J. (ed.) Computational Problems in Abstract Algebra, pp. 263–297 (1970)
7. Microsoft Security Advisory, http://www.microsoft.com/technet/security/advisory/979352.mspx
8. Operation Aurora Hit Google, Others. McAfee, Inc. (January 14, 2010)
9. CVE-1999-0256, http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-1999-0256
10. Miyachi, T., Basuki, A., Mikawa, S., Miwa, S., Chinen, K.-i., Shinoda, Y.: Educational Environment on StarBED —Case Study of SOI. In: Asia 2008 Spring Global E-Workshop–, Asian Internet Engineering Conference (AINTEC) 2008, Bangkok, Thailand, pp. 27–36. ACM (November 2008) ISBN: 978-1-60558-127-9
11. Ando, R.: Automated Log Analysis of Infected Windows OS Using Mechanized Reasoning. In: 16th International Conference on Neural Information Processing ICONIP 2009, Bangkok, Thailand, December 1-5 (2009)
12. Schneider, S., Beschastnikh, I., Chernyak, S., Ernst, M.D., Brun, Y.: Synoptic: Summarizing system logs with refinement. Appeared at the Workshop on Managing Systems via Log Analysis and Machine Learning Techniques, SLAML (2010)

Fast Protocol Recognition by Network Packet Inspection

Chuantong Chen, Fengyu Wang, Fengbo Lin, Shanqing Guo, and Bin Gong

School of Computer Science and Technology, Shandong University, Jinan, P.R. China
chenchuantongnb@163.com,
{wangfengyu,linfb,guoshanqing,gb}@sdu.edu.cn

Abstract. Deep packet inspection at high speed has become extremely important due to its applications in network services. In deep packet inspection applications, regular expressions have gradually taken the place of explicit string patterns for its powerful expression ability. Unfortunately, the requirements of memory space and bandwidth using traditional methods are prohibitively high. In this paper, we propose a novel scheme of deep packet inspection based on non-uniform distribution of network traffic. The new scheme separates a set of regular expressions into several groups with different priorities and compiles the groups attaching different priorities with different methods. When matching, the scanning sequence of rules is consistent with their priorities. The experiment results show that the proposed protocol recognition performs 10 to 30 times faster than the traditional NFA-based approach and hold a reasonable memory requirement.

Keywords: Distribution of network traffic, Matching priority, Hybrid-FA, DPI.

1 Introduction

Nowadays, identification of network streams is an important technology in network security. Due to the reason that the traditional port-based application-level protocol identification method is becoming much more inaccurate, signature-based deep packet inspection has take root as a useful traffic scanning mechanism in networking devices. In recent years, regular expressions are chosen as the pattern matching language in packet scanning applications for their increased expressiveness. Many content inspection engines have recently migrated to regular expressions.

Finite automata are a natural formalism for regular expressions. There are two main categories: Deterministic Finite Automaton (DFA) and Nondeterministic Finite Automaton (NFA). DFAs have tempting advantages. Firstly, DFAs have a foreseeable memory bandwidth requirement. As is well known, matching an input string involves only one DFA state scanning per character. However, DFAs corresponding to large sets of regular expressions can be prohibitively large. As an alternative, we can try non-deterministic finite automata (NFA) [1]. While the NFA-based representation can reduce the demand of the memory, it may result in a variable, maybe very large, memory bandwidth requirement.

Unfortunately, the requirement, reasonable memory space and bandwidth, cannot be met in many existing payload scanning implementations. To meet the requirement,

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 37–44, 2011.
© Springer-Verlag Berlin Heidelberg 2011

in this paper we introduce an algorithm to speed up the scanning of stream payload and hold a reasonable memory requirement based on the network character below.

a) The distribution of network traffic is not the same in every network. In a network, the distribution of application-layer protocol concentration is highly non-uniform, a very small number of protocols have a very high concentration of streams and packets, whereas the majority of protocols carry few packets.

According to the non uniform distribution of network traffic, we segregate the regular expressions into multiple groups and assign different priorities to these groups. We translate the regular expressions in different priority groups into different Finite Automata forms. This way proposed can achieve good matching effect.

2 Background and Relate Work

It is imperative that regular expression matching over the packet payload keep up with the line-speed packet header processing. However, this can't be met in the exiting deep packet matching implementations. For example, in Linux L7-filter, when 70 protocol filters are enabled, the throughput drops to less than 10Mbps, seriously below the backbone network speed. Moreover, over 90% of the CPU time is spent in regular expression matching, leaving little time for other intrusion detection [4].

While the matching speeds of DFAs are fast, the DFAs corresponding to large sets of regular expressions can be prohibitively large in terms of numbers of states and transitions [3]. For this many attentions are attracted to reduce the DFA's memory size. Since a explosion of states can occur when a large set of rules are grouped together into a single DFA, Yu et al. [4] have proposed segregating rules into multiple groups and building the corresponding Finite Automates respectively. Kumar et al. [2] observed that in DFAs many states have similar sets of outgoing transitions. Then they introduced Delayed Input DFA (D2FA). The D2FA is constructed by transforming a DFA via incrementally replacing several transitions of the automaton with a single default transition. Michela Becchi [9] proposed a hybrid DFA-NFA Finite Automaton (Hybrid-FA), a solution bringing together the strengths of both DFAs and NFAs. When constructing a hybrid-FA, any nodes that would contribute to state explosion retain an NFA encoding, while the rest are transformed into DFA nodes. The result is a data structure with size nearly that of an NFA, but with the predictable and small memory bandwidth requirements of a DFA. In this paper we will adopt this method in our algorithm to realize some regular expressions.

The methods above just only about how to realize the regular expressions in forms. They didn't analyze the application domain features of the regular expressions. There is no single truth what its application feature looks like. It depends on where it used. We can make use of these application characteristics when matching. We found that non-uniform distribution of packets or streams among application-layer protocols is a widespread phenomenon at various locations of the Internet. This suggests that a high-speed matching engine with small-scale storage requirement just as cache in memory system can be employed to improve performance. In this paper we propose

an algorithm to speed up the packet payload matching based on the non-uniform distribution characteristics of network streams.

3 Motivation

As is well known, the distribution of application-layer data streams is highly non-uniform; very few application layer protocols account for most of the network flows. This is called the mice and elephant phenomenon.

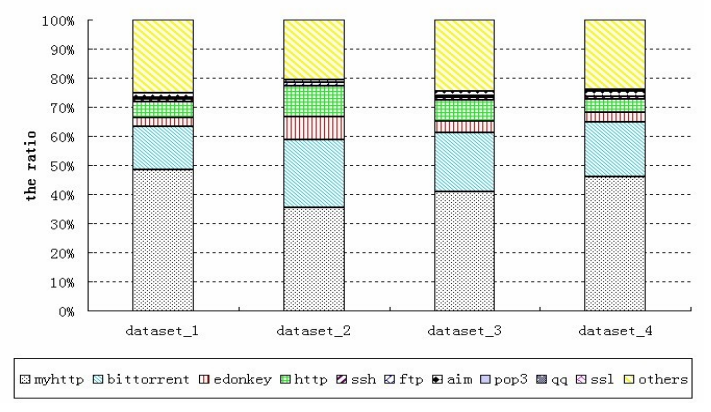


Fig. 1. Distribution of Protocol streams in different time

In Fig.1, four datasets are sampled from the same network in different time. The Figure 1 shows that the distribution of each protocol streams in a network doesn't keep stabile all the time. For example, the ratio of edonkey streams in dataset_1 is about 2% while it rises near to 7% in dataset_2. But the distribution characteristics of network streams are obvious that large fraction of the transport layer streams are occupied by very little application layer protocols. We can see that the four main application layer protocols (Myhttp, BitTorrent, edonkey, http) almost account for 70% transport layer streams in dataset_1 and even about 80% in dataset_2, the majority of infrequently used protocols ("others" in the figure 1) account for small fraction in the network. Fortunately, the non-uniform traffic distribution suggests that when matching, if we assign high-priority to the protocols which are used heavily in the network and low-priority to the lightly ones, the matching speed would be accelerated. Therefore, when we inspecting network packets, we have no excuse to ignore distribution features of the network streams scanned. However, in many existing payload scanning implementations they did not focus on the characteristics of traffic distributions in a network. In these implementations, so many unnecessary matches are tried that the matching speech is very low. In this paper we will speed up the matching based on the non-uniform characteristics of the network. We assign different priorities to protocols according to the ratio of each in network traffic and compile different priority protocol rules with different methods.

4 Proposed Method

While the distribution characteristics of network streams are useful in deep packet inspection, we introduced an algorithm to accelerate the matching speed based on the non-uniform distribution characteristics of network streams.

According to the algorithm we sample network traffic datasets firstly. These datasets can reflect the distribution of network traffic. Then we calculate the traffic ratio of every protocol in the sample dataset. We match the heavy protocol regular expressions firstly and the light ones lastly. Just as the Maximum likelihood theory, when matching we will consider the unknown network stream transmit the traffic of maximum ratio protocol, and then match it against the maximum ratio protocol rules firstly. To the few protocol rules with high priority, a high-speed matching engine of small-scale storage requirement just as cache can improve performance.

Complete description of algorithm

(1) **INPUT:** choose regular expressions → ***set A;***
(2) sort the rules in ***set A*** according to the ratio → ***set B***;
(3) ***For*** (i=0;i<B.length();i++)
(4) ***If*** the DFA states of rule B[i] <=***B0*** && ratio(B[i])> ***C0***
(5) rule B[i] → **set DFA**;
(6) ***Else*** rule B[i] → ***set NODFA***;
(7) ***End for*** ;
(8) 1→ ***i*** ;
(9) ***{ Ø }→ group_i***;
(10) ***While (!set DFA.empty( ))***
(11) ***If*** DFA states of (***group_i*** + the first rule of ***set DFA***) <=***B0***
(12) Move the first rule of set DFA → ***group_i*** ;
(13) ***Else*** ***i*** ++;
(14) ***End while;***
(15) divide rules in ***set NO_DFA →gro_1,gro_2,….gro_E0*** make the rules in each group average;
(16) **OUTPUT:** list ***group_1,group_2,…, gro_1,gro_2,…gro_E0***, into a engine;

Firstly, choose rules which we interested in or that does not conform to the network's stated security policy. For example, we can choose Thunder, edonkey or other p2p protocols. We also can select application-layer protocols be transmitted only by UDP steams or by TCP streams.

We sample one or more datasets of the network traffic from the edge router, then analyze the traffic datasets using exiting implementations, and finally compute the ratio of each chosen protocol in every examined dataset. The result of the analyses reflects the network's composition. We even can not sample datasets, just supposing the composition of the network traffic experientially. But this way may be imprecise.

Using the protocol rules chosen, we build a matching system based on the non-uniform distribution characters in network. First, rank the protocol regular expressions in accordance with the traffic scale of each protocol in the sample dataset. The

heavy protocol regular expressions locate in the front of the matching list and the light ones in the tail. For each rule, try to build a DFA. If states of the DFA are more than B0 or the traffic ratio of the corresponding rule less than C0 then move the regular expression into NO-DFA set, otherwise move the regular expression into DFA set. In both rule sets, the regular expressions are also ranked according to the ratio, using the method mentioned above. We divide the rules in DFA set into groups just like the method introduced by Yu [4]. Divide the rules into groups in accordance with the sequence of rules in DFA set. When dividing the rules into groups according to the sequence of rules in dataset, make sure that the rules in each group are as many as possible but the states of the DFA representing the group rules are not more than C0. To the NO-DFA set, we divide the rules into E0 groups, in which the rules are translated into a hybrid finite automaton (Hybrid-FA). The principle to divide the rules in NO-DFA set is just simply according to the number of rules. Make sure that the E0 is not too large and the states in each finite automaton are not too many.

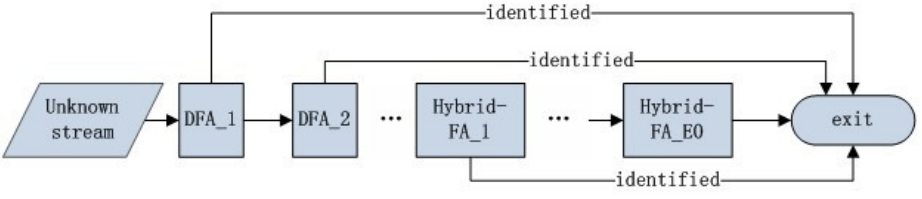


Fig. 2. Matching sequence of unknown streams

Translate the rules in each DFA group into a DFA, the rules in each NO-DFA group into a Hybrid–FA. We build the matching system just as the structure displayed in Fig.2. When scanning an unknown stream, the system will match against the rules in DFA group_1 firstly. If the first DFA DFA_1 (it corresponds to the rules in DFA group_1) can't recognize the stream then the system will activate the second DFA. If the system also can't recognize the stream, the third one… or even the last Hybrid-FA maybe used. For the non-uniform distribution of network, many streams can be recognized through only a few matching engines.

6 Evaluation Result

Select application-layer protocol rules, transmitted by the TCP streams, from the L7-filter rules set. The rule, myhttp, is a new description pattern of http protocol. Its performance is good. We choose 50 TCP rules from the rules set totally and group the rules according to our algorithm. In this example experiment, the parameter B0=1000, C0=1% and E0=4 .The result of grouping are displayed in table 1.

Our network traffic datasets were obtained from the network of High Performance Computing Center in Shandong University. The size of each dataset is not more than 1G, in which the size of each packet is not more than 1500 bytes. Select one dataset as a sample and analyze its composition. We record stream numbers of each protocol in

total and compute the traffic ratio of each in the dataset. We build the matching system according to our algorithm using the chosen regular expressions.

Table 1. Results of Grouping

DFA	Group_1	bittorrent http myhttp
	Group_2	aim ftp qq ssh
	Group_3	pop3 ssl telnet irc
	Group_4	smb socks cvs msnmessenger
NO-DFA	Gro_1	edonkey aimwebcontent applejuice ares bgp chikka cimd ciscovpn
	Gro_2	counterstrike-source directconnect fasttrack imap gkrellm gnucleuslan gopher hddtenpt ident
	Gro_3	ipp jabber msn-filetransfer napster nntp openft radmin rdp rlogin
	Gro_4	rtsp sip smtp soulseek subversion vnc yahoo xunlei unknown

Our goal is to compare the matching speed of our matching system with the NFAs. We also compare it with the matching system which is constructed in a similar way with our algorithm but the NO-DFA set groups are built by NFAs. The matching speed is measured with the number of the states the matching system visit, the more states are visited the slower the matching speed is. Meanwhile, we will compare each system's state scale.

We examine the performance of different matching systems with simulated experiment. Each system scans the same datasets we sampled. In each, analyze the number of states traveled in total and record stream numbers of identified as every protocol. We define a parameter SR (Speed-up ratio) to evaluate our system's performance. TO see how, consider the travel speed v between states is the same in each matching system. When matching the dataset θ, the scanning time of matching system α is t_α, the number of states visit is S_α. While in matching system β, when it matches the dataset θ, the scanning time is t_β, the number of states visit is S_β.

$$t_a = \frac{S_\alpha}{v} \tag{1}$$

$$t_\beta = \frac{S_\beta}{v} \tag{2}$$

The speed up ratio of matching system β can be computed as follow:

$$SR = \frac{t_\alpha}{t_\beta} \tag{3}$$

Combining (1) and (2), the following expression can be obtained:

$$SR = \frac{S_\alpha}{S_\beta} \tag{4}$$

Table 2 displays the average flows' number of classified as protocols list by each classifier. For each application layer protocol, the numbers of streams identified by the matching systems are almost the same. But the states visited (showed in table 3) in the simplex NFA system are nearly more than 30 times of our algorithm. That is to say the matching speed of our system is nearly 30 times faster than the simplex NFA system. Obviously, our matching algorithm based on the distribution characteristics of network traffic is very effective. In the memory scale aspect, although the total states in our system increase by many times, they are still not more than 30000. This scale is acceptable completely. Comparing our system with the similar system (DFA-NFAs system), which also groups the regular expressions like our algorithm but the NO-DFA set groups are built by NFAs, we can see that the states visited in our system reduced by more than 4 times. That's because many rules use multiple wildcard meta-characters (e.g., '.', '*') in the rule set, even majority of the wildcards are used with length restrictions ('?', '+'). As these regular expressions are converted into state machines for pattern matching, large numbers of wildcards and length restrictions can cause the corresponding DFA to grow exponentially. In our system we used the hybrid-FA form; it has a modest memory storage requirement comparable with it of an NFA solution, but an average case memory bandwidth requirement similar to that of a single DFA solution, a worst case memory bandwidth linear in the number of regular expressions containing counting constraints and dot-star conditions [9].

Table 2. The number of identified flows

Application Protocol	NFA	DFA-NFAs	Our system
aim	39	39	39
bittorrent	615.333	615.333	615.333
edonkey	167	165.167	165.167
ftp	16.667	16.667	16.667
http	125.833	127.667	127.667
pop3	1.833	1.833	1.833
ssh	15.667	15.667	15.667
telnet	0.167	0.167	0.167
myhttp	1087.5	1087.5	1087.5
unknown	687	687	687

Table 3. The performance of each system

	NFA	DFA-NFAs	Our system
# of state in system	1202	3375	27248
# of states visited	21071446.33	3237741.17	741905.67
reduction rate of states	0	84.63 %	96.48 %
S R	1	6.51	28.40

6 Conclusion

As a matter of fact, the distribution of application-layer protocols is non-uniformed in network that a very small number of protocols have a very high concentration of

packets and flows, whereas the majority of protocols carry very few packets. If we match against rules equally, many redundant matches will be tried and the matching speech will be very low.

Based on the distribution of network streams, regular expressions are separated into groups and assigned different matching priorities according to their distribution. Rules in a high-priority group are constructed into a high-speed matching pattern DFA and be scanned according to the priority order. For the protocol rules in a low-priority group, we translate them into a low-speed matching pattern Hybrid-FA and match against them after the DFAs. The experiments suggest that our algorithm based on the distribution of network streams is very effective. For heavy rules are matched against firstly, many redundant visits can be removed. Our algorithm can achieve 6 to 30 times speed up compare with simple NFA forms and its memory scale is acceptable.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant No. 60803142, Natural Science Foundation of Shandong Province for Youths (Grant No. Q2008G01), Independent Innovation Foundation of Shandong University (Grant No. 2009TS031), the Key Science-Technology Project of Shandong Province of Shandong (Grant No. 2010GGX10117), Open Research Fund from Key Laboratory of Computer Network and Information Integration In Southeast University, Ministry of Education.

References

1. Sidhu, R., Prasanna, V.K.: Fast Regular Expression Matching using FPGAs. In: The 9th Annual IEEE Symposium on FCCM (2001)
2. Kumar, S., Dharmapurikar, S., Yu, F., Crowley, P., Turner, J.: Algorithms to Accelerate Multiple Regular Expressions Matching for Deep Packet Inspection. In: ACM SIGCOMM, Pisa (2006)
3. Becchi, M., Crowley, P.: An Improved Algorithm to Accelerate Regular Expression Evaluation. In: ANCS, Orlando (2007)
4. Yu, F., Chen, Z., Diao, Y., Lakshman, T.V., Katz, R.H.: Fast and Memory-Efficient Regular Expression Matching for Deep Packet Inspection. In: ANCS, California (2006)
5. Qian, X., Yue-Peng, E., Ge, J.-G., Qian, H.-L.: Efficient Regular Expression Compression Algorithm for Deep Packet Inspection. Journal of Software 20(08), 2214–2226 (2009)
6. Huiping, F., Lei, X., Shuhui, C., Gaoping, H.: Speed Up on Application Protocol Recognition Using Regular Expression. Journal of Computer Research and Development 45(Suppl.), 438–443 (2008)
7. Fang, W., Peter, L.: Inter-AS Traffic Patterns and Their Implications. In: IEEE Global Telecommunications Conference, vol. 3, pp. 1859–1868 (1999)
8. Levandoski, J., Sommer, E., Strait, M.: Application Layer Packet Classifier for Linux, `http://l7-filter.sourceforge.net/`
9. Becchi, M., Crowley, P.: A Hybrid Finite Automaton for Practical Deep Packet Inspection. In: coNEXT, New York (December 2007)
10. Bro Intrusion Detection System, `http://bro-ids.org/Overview.html`
11. Snort Network Intrusion Detection System, `http://www.snort.org`

Network Flow Classification Based on the Rhythm of Packets

Liangxiong Li[1], Fengyu Wang[1], Tao Ban[2], Shanqing Guo[1], and Bin Gong[1]

[1] School of Computer Science and Technology, Shandong University, Jinan, P.R. China
liliangxiong@126.com, {wangfengyu,guoshanqing,gb}@sdu.edu.cn
[2] Cybersecurity Laboratory, Network Security Research Institute, Tokyo, Japan
bantao@nict.go.jp

Abstract. Accurate traffic classification is a necessary means of network management, QOS, monitoring and so on. We find that each protocol's flows have their own packet-level rhythm on the statistical characteristics. In this paper we present a Bayesian network classification mechanism based on the flows' packet-level rhythm. However, the flows rhythm is always too scattered to bring into play its ability well in the Bayesian network, so we employ an Equal-width discretization method to centralize the rhythm and discretize the packet size and interval-time to some different space. Then we applied our classification model to the different discretization data set of HTTP, EDONKEY, BITTORRENT, FTP and AIM. Experiment results show that our approach can achieve better precision and recall rate for these applications.

Keywords: Network traffic classification, Bayesian network, Flow rhythm.

1 Introduction

Today, various new applications and unknown protocols make network increasingly complicated, diverse and hard to manage. For example: P2P, Video stream, VoIP and so on occupy many bandwidth, causing the network bandwidth exhaustion [1]; various network malicious attacks, such as network intrusion [2], are severely harm to network services and information security. So today, the classification of Network flows becomes the basic and effective means. It is widely applied in the application of QOS, security, monitoring, intrusion-detection, virtual private networks (VPN) and packet filtering applications [3].

Generally there are three main categories methods for such work. The simplest method is relied on the use of transport layer port numbers [4]. The second approach is relied on the detailed analysis of each packet payload according to the known applications features [4, 5]. The third approach is relied on the flow packets-level statistical characterization, such as the packet size, interval-time, etc.

Here we also present an effective modeling approach relying on a packet-level view of Internet traffic. It's true that each protocol's flows have their own rhythm, so we should find a method which can utilize the packet rhythm well. In this paper we use Bayesian network, a kind of machine learning methods, in here whose net structure is assigned by ourselves according to the correlation of feature attributes, to classify the

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 45–52, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Internet flows with at most 14 feature attributes – the first seven packet sizes and the first seven packet interval-times of a flow. However the flows rhythm is always too scattered, making the Bayesian net nodes' values distribution too big, so it will affect the precision of classifier. Therefore we need to extract the each protocol flows' rhythm accurately and make them useful and simple. We find data discretization can centralize the flows rhythm well. We will discretize the packet size and interval-time to different spaces in order to find an optimal centralization of the flow rhythm.

2 Motivation and Related Work

In nowadays, the network has brought our society huge profits, but this also has brought new challenges. How to accurately classify network flows should be the basic work to solve all these problems, however, the classification methods we used now are based on the port and packet payload, which are incompetent when faced with the current network environment. Some applications frequently run on non-predefined or other well-known ports to circumvent network policy restrictions. The classification based on packet payload is primarily through pattern matching methods to match the packet contents and existing protocol's patterns. The main drawback of such approaches is requires a lot of calculation and dependent on existing patterns. Fortunately, the packet size and packet interval-time can describe the every protocol own feature well. Early, in [3], Paxson et al. researched the relationship between the statistical properties of flows and the protocols. This works show that packet length, inter-arrival times and flow duration can be suitable to express the behavior of protocols. In [6] Alberto Dainotti et al. have structured a very good model using the packet-level features and HMM. In [7], Hyunchul Kim et al. have introduced several machine learning methods and analyzed the performance; though the features are not just flow's size and time interval, it still reflects the importance of the two features. In [8], Manuel Crotti et al. have presented a method which bases on statistics of the packet-level features and their order.

3 Packet-Level Feature and Bayesian Networks

In this paper, we define the IP flow as the unidirectional, ordered sequence of IP packets. A flow is composed by (N+1) IP packets from Pkt_0 to Pkt_N, where Pkt_j represents the j-th IP packet sent by the client to the server or by the server to client. For each packet, what we concern is just the packet size and the interval-time between itself and previous packet. So IP flow can be characterized as an ordered sequence of N pairs $P_i = \{S_i, \Delta T_i\}$, with $1 \le i \le N$, where S_i represents the size of Pkt_i and ΔT_i represents the interval-time between Pkt_{i-1} and Pkt_i [8].

3.1 Network Flow Rhythm

Through much existed work about the packet-level statistical features of the application protocols, there is a fact that the different application protocols have their own feature, which we call the flow's packet rhythm. For example, Fig. 1 show three

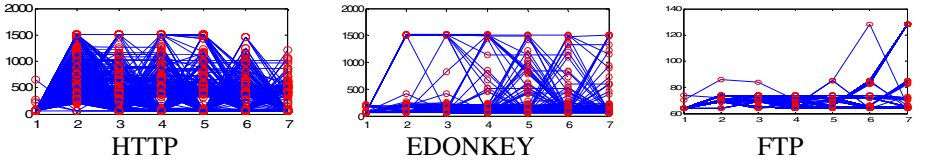


Fig. 1. Flow Packet Size Sequence

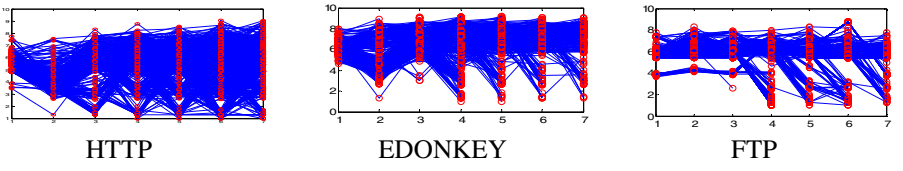


Fig. 2. Flow Packet Interval-time Sequence

protocols' packet size sequences and Fig. 2 shows these protocols' packet interval-time sequences; the two figures obviously reflect the distinction between different protocol's rhythms. In fact, the rhythm always is not obvious because the distribution of packets size and interval-time are not centralized enough.

3.2 Attributions Selection and Attributions Correlation

In this paper we only select the first seven packets size and interval-time of every flow at the most, if a flows with less than seven packets, we will select all of the packets of it. We find the rhythm is always shown through the first several packets and in [9] the author also indicates the first several packets is a good predictor for classification.

The attributions correlation with each other is the basic for our Bayesian network structure setting up. From the existed work [6, 10], we illustrate following principle.

1. Packets size and interval-time relying on network protocols;
2. Packets size and interval-time being mutual independence;
3. Packets size only relying on its front adjacent packet's size;
4. Interval-time only relying on its front adjacent interval-time.

3.3 Bayesian Networks Parameters Estimation

Considering a Bayesian network with m+1 attributes $(C^i, A_1^i, A_2^i, ..., A_m^i)$, we suppose A_i has r_i kinds of values $1,2,\cdots,r_i$ and its parents nodes have q_i kinds of combinations. Then we can define the parameters θ is as following:

$$\theta_{ijk} = P(X_i = k \,|\, \pi(X_i) = j) \tag{1}$$

Where i represents the A_i, k represents the A_i's value is equal to k and j represents the A_i's parents nodes' combination values are j. The estimation formula is as following:

$$\theta_{ijk} = \begin{cases} \dfrac{m_{ijk}}{\sum_{k=1}^{r_i} m_{ijk}} & \text{if} \quad \sum_{k=1}^{r_i} m_{ijk} > 0 \\ \dfrac{1}{r_i} & \text{if} \quad \sum_{k=1}^{r_i} m_{ijk} = 0 \end{cases} \tag{2}$$

Where m_{ijk} represents the total number of sample at the node A_i whose value is k and its parents nodes combination values are j.

3.4 Our Bayesian Network Structure

Generally, the Bayesian network structure is obtained through applying a certain algorithm to a train set, but in this case, the structure largely dependents on the train sets; so if train sets is different, the result structure might be different. The attributions correlations are not always reflected in the net structure.

In this paper, we emphasize that the Bayesian net structure is structured according to the four principles in 3.2, seeing Fig. 3. In the network structure, the root node represents the class attribution and others 14 nodes respectively represent the first seven packet size and interval-time. From the Fig. 3, we can see root node has no parents' node; the first packet size node and the first packet interval-time node only have one parents' node and all other nodes have two parent nodes.

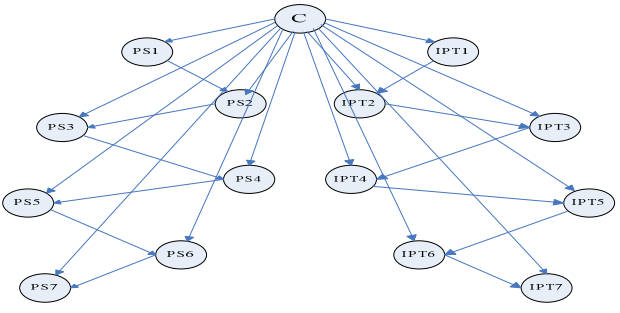


Fig. 3. Bayesian net structure

4 Experiment Setup

4.1 Data Set

Our experiments traffic traces data are collected at the edge gateway of ourselves college's campus network, using the well-known tool –TCPDUMP. Firstly, we use a classifier tool based on the L-7 filter to process the PCAP file and signature the flows with corresponding protocol, then according to the results we extract the flows packets size and interval-time to compose the train and test samples set. It is important to emphasize that we only choose the protocols which use the TCP protocol to set up

connections and have a big percentage in our PCAP files; they are respective HTTP, EDONKEY, BITTORRENT, FTP and AIM (AOL Instant Messenger).

In this paper we only take into account the client to server flows. And the composition of train and test sets is as following Table I :

Table 1. Train and Test set sizes

	HTTP	EDONKEY	BITTORRRENT	FTP	AIM
Train set	16109	1382	8905	1175	711
Test set	12130	1465	8053	857	655

4.2 Data Pre-processing

Because on an Ethernet link, the packet size would range from 64 to 1518 bytes and interval-time would range continuously from 10^{-6} to 10^{3} seconds [8], it needs a lot of space to save the states. Besides the distribution space is too scatter to centralize the flows rhythm, result in reducing the performance of the Bayesian network.

We employ a method based on Equal-width discretization [11] as following:

1. Calculating the ratio vector R of adjacent packets size of a flow, the same to the interval-time.
2. Normalize the original size or interval-time value x_i to x_i' according to the formula:

$$x_i' = \frac{x_i - \min}{\max - \min}(new_\max - new_\min) + new_\min \tag{3}$$

3. In the new space [new_min, new_max], if a data value x_i' not locates in the center section of two integrates, round its value directly, else we will select P_i between $\text{int}(x_i')$ and $\text{int}(x_i')+1$ with making the (4) smaller.

$$\frac{P_i}{P_{i-1}} - r_i \tag{4}$$

Where P_{i-1} is the discretized value of x'_{i-1}, r_i represents the ratio of x'_i and x'_{i-1}.

Table II is the discretization scale of size and interval-time:

Table 2. Discretization scale

Size	1-15	1-20	1-25	1-30
Interval-time	1-6	1-10	1-15	1-20

5 Result and Analysis

Finally, we find when the size is discretized to the space from 1 to 25 and the interval-time is discretized to the space from 1 to 10, the flow rhythm can be best centralized

and the classification result is optimal. We have to define two parameters, one is the precision rate **PT** and the other is the recall rate **RT**. We define $\mathbf{E^P}$ as the flows number of protocol p in the evaluation set, define $\mathbf{C^P}$ as the total flow number of classified as protocol p by our classifier and $\mathbf{T^P}$ as the flows number of correctly classified as protocol p by our classifier.

$$PT = \frac{T^P}{C^P} \tag{5}$$

$$RT = \frac{T^P}{E^P} \tag{6}$$

We also have make a experiment with no any pre-process the data sets just except binning the interval-time with step 0.01, so that we can compare the classification results between the centralization data and original data. The Table III shows the optimal classification Confusion matrix. The Fig. 4 shows the PT of the centralization data and original data, and the Fig. 5 shows the RT of the centralization data and original data. It is very clearly that the centralization data have a good classification result with more than 94% precision and recall rate, so it can illustrate that the centralization could centralize the flows rhythm well and the centralizing rhythms are more value to the flows classification.

Table 3. Optimal classification Confusion matrix

	HTTP	EDONKEY	BITTORRENT	FTP	AIM		
						PT	RT
HTTP	12054	41	28	7	0	99.6%	99.4%
EDONKEY	31	1387	47	0	0	96.4%	94.7%
BITTORRENT	12	10	8030	1	0	99.0%	99.7%
FTP	1	1	5	850	0	98.5%	99.2%
AIM	0	0	3	5	647	100%	98.8%

The Fig.6 and Fig.7 show the change of PT and RT with different packet size discretization scale and fixed interval-time discretization scale 1-10. We can see if the discretization scale is too small, it is over-centralization for the rhythm; so the PT and RT will both decrease greatly and for some protocols may be worse, such as EDONKEY. But as the discretization scale becomes increase, the PT and RT won't increase, instead it may decrease, because the rhythms begin to become scattered. Finally, we find when the size is discretized to the space from 1 to 25 is the optimal.

The Fig.8 and Fig.9 show the change of PT and RT with different packet interval-time discretization and fixed size discretization scale 1-25. It is also existed over-centralization of the rhythm, so the PT and RT will decrease too. As the discretization scale becomes increase, the PT and RT won't increase, it also may decrease owing to the rhythm beginning to become scattered. Finally, we find when the interval-time is discretized to the space from 1 to 10 is the optimal.

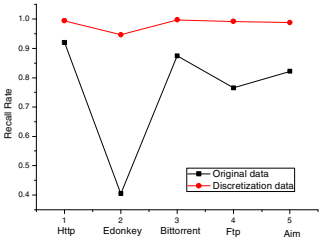


Fig. 4. Precision Rate

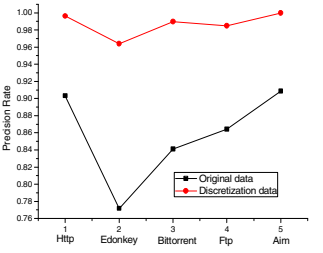


Fig. 5. Recall Rate

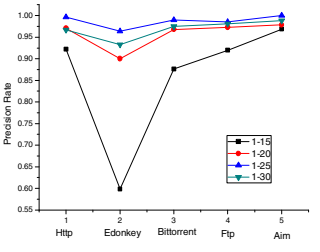


Fig. 6. PT vs. PS discretization

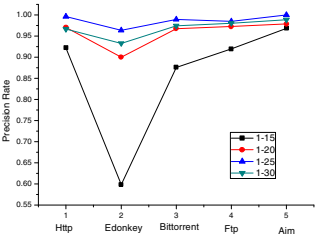


Fig. 7. RT vs. PS discretization

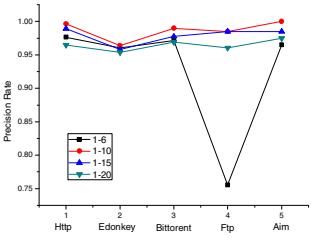


Fig. 8. PT vs. IPT discretization

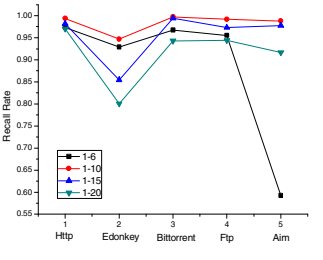


Fig. 9. RT vs. IPT discretization

Besides, as the net nodes values distribution becomes bigger and bigger, it will lead to the Bayesian network parameters increasing and cause the training of Bayesian net becoming more complexity; meanwhile it will cost more memory space to store the net parameters.

6 Conclusion

In this paper we proposed a Bayesian network classification mechanism based on the flow packets-rhythm. We demonstrated the rhythm's value in experiments using the

first seven packets size and interval-time. We compared the precision rate and recall rate of discretization data and original data and find the centralization of flow rhythm shows a better performance. Furthermore, in order to find an optimal centralization of rhythm, we discretize the packet sizes and interval-time to many different spaces in experiments, and we find the optimal centralization finally.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China under Grant No. 60803142, Natural Science Foundation of Shandong Province for Youths (Grant No. Q2008G01), Independent Innovation Foundation of Shandong University (Grant No. 2009TS031), the Key Science-Technology Project of Shandong Province of Shandong (Grant No. 2010GGX10117), Open Research Fund from Key Laboratory of Computer Network and Information.

References

1. Zander, S., Nguyen, T., Armitage, G.: Automated Traffic Classification and Application Identification using Machine Learning. In: The IEEE Conference on Local Computer Networks 30th Anniversary (LCN 2005), pp. 250–257 (2005)
2. Li, W., Moore, A.W.: A Machine Learning Approach for Efficient Traffic Classification. In: 2007 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, pp. 310–317 (2007)
3. Paxson, V., Floyd, S.: Wide area traffic: the failure of Poisson modeling. IEEE/ACM Trans. Netw., 226–244 (1995)
4. Crotti, M., Dusi, M., Gringoli, F., Salgarelli, L.: Traffic Classification through Simple Statistical Fingerprinting. ACM SIGCOMM Computer Communication Review 37, 5–16 (2007)
5. Karagiannis, T., Papagiannaki, K., Faloutsos, M.: BLINC: Multilevel Traffic Classification in the Dark. ACM SIGCOMM Computer Communication Review 35(4), 229–240 (2005)
6. Dainotti, A., Donato, W.D., Pescapè, A., Rossi, P.S.: Classification of Network Traffic via Packet-Level Hidden Markov Models. In: Global Telecommunications Conference, IEEE GLOBECOM, pp. 2138–2142 (2008)
7. Kim, H., Claffy, K., Fomenkov, M., Barman, D., Faloutsos, M., Lee, K.Y.: Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices. In: Proceedings of the 2008 ACM CoNEXT Conference (2008)
8. Moore, A.W., Zuev, D.: Internet Traffic Classification Using Bayesian Analysis Techniques. In: Proceedings of ACM SIGMETRICS, Banff, Canada (2005)
9. Bernaille, L., Teixeira, R., Akodkenou, I., Soule, A., Salamatian, K.: Traffic Classification on the Fly. ACM SIGCOMM Computer Communication Review 36
10. Wright, C., Monrose, F., Masson, G.M.: HMM Profiles for Network Traffic Classification. In: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, pp. 9–15
11. Yang, Y., Webb, G.I., Wu, X.D.: Discretization Methods. Data Mining and Knowledge Discovery Handbook, 2nd edn., pp. 101–115 (2010)

Energy-Based Feature Selection and Its Ensemble Version

Yun Li and Su-Yan Gao

College of Computer and Institute of Computer Technology
Nanjing University of Posts and Telecommunications
No.66, Xinmofan Rd., Nanjing 210003, China
liyun@njupt.edu.cn

Abstract. Variable and feature selection has been a research topic with practical significance in many areas such as statistics, pattern recognition, machine learning and data mining. The task of feature selection is to choose an effective feature subset out of a given feature set to reduce the feature space dimensionality. In this paper, along with the guidelines of Energy-based model, a unified energy-based framework for feature selection and a feature ranking algorithm under this framework is presented. On the other hand, in order to increase the stability of our algorithm, an ensemble feature selection is introduced. Some experiments are conducted on the real world and synthesis data sets to demonstrate the ability of our feature selection algorithm and the stability improvement of the ensemble feature selection.

Keywords: Feature Selection, Loss, Energy-Based Model, Ensemble.

1 Introduction

Data mining as a multidisciplinary joint effort from databases, machine learning, and statistics, is championing in turning mountains of data into nuggets [1]. Researchers and practitioners realize that in order to use data mining tools effectively, data preprocessing is essential for successful data mining. Feature selection is one of the important and frequently used techniques in data preprocessing for data mining. In general, The data is represented by a very large number of features, but only few of them are relevant for predicting the label. In addition, many algorithms become computationally intractable when the feature dimension is high. Therefore feature selection, i.e. the task of choosing a small subset of features which is sufficient to predict the target labels well, is critical to minimize the classification error. At the same time, feature selection can also reduce system complexity, processing time of data analysis and the cost of collecting irrelevant features.

Roughly speaking, feature selection algorithms have two key problems: search strategy and evaluation criterion. According to the criterion, algorithms can be categorized into three classes: the filter model, the wrapper model and the embedded model [1]. In the wrapper model the selection method tries to directly

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 53–62, 2011.
© Springer-Verlag Berlin Heidelberg 2011

optimize the performance of a specific predictor (algorithm). The main drawback of this method is its computational deficiency. In the filter model the selection is done as a preprocessing, without trying to optimize the performance of any specific predictor directly. This is usually achieved through an (ad-hoc) evaluation function using a search method to select a feature subset that maximizes this function. To take advantage of the above two models and avoid the prespecification of a stopping criterion, the embedded model is presented. Different methods apply a variety of search heuristics, such as hill climbing and genetic algorithms. References [1,2] have given a comprehensive discussion of feature selection methodologies.

As the extension of statistical learning theory, the energy-based model [3] can avoid the probability estimation and provide a framework for many probabilistic and non-probabilistic methods in machine learning. In the paper, along with the guidelines of energy-based model, an improved energy-based framework for feature ranking is introduced, and a ranking algorithm stems from the framework is presented. Feature ranking is a filter method: it is a preprocessing step, independent of the choice of the predictor. Still, under certain independence or orthogonal assumptions, it may be optimal with respect to a given predictor. Even when feature ranking is not optimal, it may be preferable to other feature subset selection methods because of its computational and statistical scalability: computationally, it is efficient since it requires only the computation of n feature scores and sorting the scores. Statistically, it is robust against over-fitting because it introduces bias but it may have considerably less variance [2]. On the other hand, the stability of feature selection has been attracted much attention [4,5,6]. Then stable ensemble version of our proposed algorithm is presented.

The paper is organized as follows: section 2 presents energy-based frameworks for feature ranking, and a feature ranking algorithm is introduced in section 3. The ensemble feature selection algorithm is shown in section 4. The experimental results are shown in section 5, and the paper concludes in section 6.

2 Energy-Based Framework for Feature Ranking

2.1 Energy-Based Learning

Energy-based learning [3] provides a unified framework for many probabilistic and non-probabilistic approaches to learning. Let us assume the training set S contains N samples, $\{X, Y\} = \{x_i, y_i\}_{i=1}^{N}$, and each sample is represented by an n-dimensional vector $x_i = (x_{i1}, x_{i2}, \cdots, x_{in}) \in \mathcal{R}^n$ and discrete class labels y_i. A energy function is used to measure the conformity of each possible configuration of X and Y, whose value can be interpreted as the degree of compatibility between X and Y. For convention, small energy values correspond to highly compatible configurations, while large energy values mean highly incompatible configurations of X and Y. The energy function is denoted as $E(Y, X)$. In the most common use of a model, the input X is given (observed from the world), and the model produces the answer Y that is most compatible with the observed X. Then an Energy-Based Model should be trained to find an energy function

that produces the most compatible Y for any X. The best energy function is searched within a family of energy functions β indexed by a parameter w.

$$\beta = \{E(w, Y, X) : w \in W\} \tag{1}$$

In order to find the best energy function, we need to assess the quality of any particular energy function, only based on the training set S and our prior knowledge about the task. This quality measure is called the loss function and denoted by $L(E, S)$. For simplicity, we often denote it as $L(w, S)$. Then the learning problem is simply to find the w' that minimizes the loss:

$$w' = argminL(w, S) \tag{2}$$

2.2 Framework for Feature Ranking

It is evident that the loss function for a pair sample should be designed in such a way that it assigns a low loss to well-behaved energy functions: energy functions that give the lowest energy to the correct answer and higher energy to all other (incorrect) answers. Conversely, energy functions that do not assign the lowest energy to the correct answers would have a high loss [3].

Definition 1. Let Y be discrete variable, then for a training sample (x_i, y_i), *the most offending incorrect answer* is the answer that has the lowest energy among all answers that are incorrect:

$$\overline{y}_i = argmin_{y \in Y, y \neq y_i} E(w, Y, x_i) \tag{3}$$

With the properly designed loss function, the learning process should reduce the energy of $E(w, x_i, y_i)$ and increase the energy of incorrect answers, particularly on $E(w, \overline{y}_i, x_i)$. Different loss functions do this in different ways. The generalized margin loss directly uses the energy of the most offending incorrect answer in the contrastive term to create an energy gap between the correct answer and the incorrect answers [3]:

$$L_S(w, x_i) = Q_m(E(w, y_i, x_i), E(w, \overline{y}_i, x_i)) \tag{4}$$

As introduced in [3], where m is a positive parameter called the margin and $Q_m(e_1, e_2)$ is a convex function whose gradient has a positive dot product with the vector [1,-1] in the region where $E(w, y_i, x_i) + m > E(w, \overline{y}_i, x_i)$. In other words, the loss surface is slanted toward low values of $E(w, y_i, x_i)$ and high values of $E(w, \overline{y}_i, x_i)$ wherever $E(w, y_i, x_i)$ is not smaller than $E(w, \overline{y}_i, x_i)$ by at least m. Many loss functions, such as hinge loss, log loss, square-square loss and square-exponential loss, etc, can be used in the (4).

The equation above only consider one most offending incorrect answer, which is sensitive to noise and outlier. Then an improved framework is proposed with the consideration of several target neighbors for a sample.

Definition 2. *Target Neighbors*, k nearest neighbors with the same label to sample x_i are named as target neighbors of x_i [7,8].

Definition 3. *Target Matrix* $\boldsymbol{T}$, whose element $t_{ij} \in \{0,1\}(i = 1, 2, \cdots, N, j = 1, 2, \cdots, N)$ indicates whether x_j is a target neighbor of x_i. $t_{ij} = 1$ denotes x_j is a target neighbor of x_i. A sample is a target neighbor of itself $t_{ii} = 1$.

Then the energy framework with k target neighbors is proposed as follows.

$$L_S(w, x_i) = \sum_{t_{ij}=1} Q_{m_i}(E(w, y_j, x_j), E(w, \overline{y}_j(m_i), x_j)) \tag{5}$$

$\overline{y}_j(m_i)$ denotes the incorrect answers in the region with margin m_i for each target neighbor of x_i. In other words, the offending incorrect answers for each target neighbor of x_i in a predefined margin m_i are involved in the loss function of x_i. In general, m_i should be set to assure the most offending incorrect answer $\overline{y}_i$ for x_i is involved in $L_S(w, x_i)$. If the parameter w is considered as the feature weight vector, then the feature ranking is simply to find the w' that minimizes the loss function defined in (5). And the feature selection is to choose the most important features from the ranked feature set.

3 Feature Ranking Algorithm

3.1 Evaluation Function

Now, we will present a concrete feature ranking algorithm under the proposed energy-based framework above. The critical issues are the properly designing the internal structure of energy function E and the loss function L based on the key point that energy functions should give the lowest energy to the correct answer and higher energy to all other (incorrect) answers [3]. In general, in order to achieve high generalization, the target neighbors of a sample always should be closer, while other samples from different classes are separated by a large margin. Then the target neighbors can be considered as the "correct answers" of a sample, while other samples from different classes are the "incorrect answers". And the energy in this case can be defined as the weighted distance between samples, the parameter w is the feature weights and the most offending incorrect answer is the most offending neighbor, which is the closest sample to x_i with different label. In this paper, Euclidean distance is used and the energy is defined as follows.

$$E(w, x_i, x_j) = ||x_i - x_j||_w \tag{6}$$

where $||x_i - x_j||_w = \sqrt{\sum_f w_f^2 (x_{if} - x_{jf})^2}$, $w_f \in [0,1], f = 1, 2, ...n$. x_j $(j = 1, 2, \cdots, N)$ are samples in training set S contains N samples and n features, x_{if} is the f'th feature of x_i. For each input x_i, the loss function can be defined as:

Definition 4. *let S be a training set and x_i be a sample, w be a weight vector, then the loss function of x_i is*

$$L_S(w, x_i) = \sum_{t_{ij}=1} (E(w, x_i, x_j)^2 + c \log(1 + \exp^{max(0, m_i + E(w,x_i,x_j)^2 - E(w,x_i,x_p)^2)}) \tag{7}$$

x_p $(p = 1, 2, \cdots, N)$ are samples in S with different label, i.e., $y_p \neq y_i$. $c > 0$ is some positive scaling factor (typically set by cross validation). t_{ij} is the entry in Target Matrix $\boldsymbol{T}$. The loss function is derived from the framework defined in (5), and it is the combination of square loss with log loss. $L_S(w, x_i)$ is a convex function whose gradient has a positive dot product with the vector [1,-1] in the region where $E(w, x_j, x_i) + m_i > E(w, x_p, x_i)$, and it is generally convergent [3].

For the two competing terms in the proposed loss function, the first term penalizes a high energy for target neighbors, while the second term penalizes low energy for other samples that do not share the same label with x_i. In the second term, the different labelled samples located in area from x_i to any of its target neighbors plus a predefined margin m_i is concerned. m_i is a positive constant and defined as follows.

$$m_i = |||x_i - nearmiss(x_i)||^2 - ||x_i - nearhit(x_i)||^2| \tag{8}$$

where $nearhit(x_i)$ and $nearmiss(x_i)$ denotes the nearest samples to x_i with the same and different label respectively. And the $nearhit(x_i)$ and $nearmiss(x_i)$ is also the nearest target neighbor and the most offending neighbor for x_i respectively. The definition of m_i assures the most offending neighbor, i.e.,$nearmiss(x_i)$, will be involved in the loss function. Then the proposed loss function surely meets the requirements of the framework defined in (5).

The parameter w is treated as the feature weight, then Definition 4 allows weight over the features in the energy calculation. A change of weights will cause a change in the energy. At the same time, a chosen subset of features affects the loss through the energy measure. Then we can identify important features by their weights. When a training set is available, we sum the loss over the instances to obtain the evaluation criterion:

$$e(w) = \sum_i L_S(w, x_i) \tag{9}$$

3.2 Algorithm Analysis

The w is treated as the feature weight and the magnitude of each element in w measures the relevance of corresponding feature, then we can identify important features by their weights. Since loss function defined in (7) is smooth almost everywhere, we can use gradient descent to find the weight vector w. The computational time of the proposed algorithm mainly contains the calculation of pairwise energy between samples, and solving the optimization problem that minimizing the total loss function of all samples, they always have time complexity $O(N^2n)$ and $O(Nkn)$ respectively, where k is the number of target neighbors. In general, k is a small constant number. Then the total time complexity is $O(N^2n)$. Our previous work-Lmba [9] also stems from the framework above. And the proposed algorithm in this paper is the soft version of Lmba using the log loss instead of hinge loss and the proposed algorithm is named as L-Lmba.

4 Ensemble Feature Selection

For feature selection, besides high accuracy, another important issue is the stability of feature selection. Stability is the insensitivity of a feature selection algorithm to variations of the training set. This issue is particularly critical for the small-sample data with high dimensionality. In general, a feature selection algorithm often chooses largely different feature subsets because of the variations to the training data, although most of these subsets are as good as each other in terms of classification performance [4,5]. However, such instability reduces the confidence of domain experts in experimentally validating the selected features. Then the stability of a feature selection algorithm should be paid more attention. In order to improve the stability of our algorithm, the ensemble feature selection is adopted, which can improve the stability of feature selection algorithm without sacrificing classification accuracy [6].

4.1 Components of Ensemble Feature Selection

Similar to the ensemble models for supervised learning, there are two essential steps in ensemble feature selection . The first step involves creating a set of different feature selectors, each provides its output, while the second step aggregates the results of all feature selectors. In our case, variation in the feature selectors can be achieved by instance level perturbation. We adopt a subsampling based strategy. For a training set, l subsamples of size $\alpha M(0 < \alpha < 1)(M$ is the number of samples in the current training set) are drawn randomly, where the parameters α and l can be varied. Subsequently, feature selection is performed on each of the l subsamples. Aggregating the different feature selection results can be done by weighted voting.

Consider an ensemble $En = \{w_1, w_2, \cdots, w_l\}$, which consisting of l feature selectors, then each $w_g = (w_g^1, w_g^2, \cdots, w_g^n)(g = 1, 2, \cdots, l)$ is the feature weighting result of our proposed algorithm (L-Lmba). $w_g^t(t = 1, 2, \cdots, n)$ represents the feature weight of feature t for g-th feature selector. However, feature weighting is almost never directly used in ensemble feature selection [6], and instead converted to a ranking. Then we can obtain the corresponding ensemble ranking set $\{r_1, r_2, \cdots, r_l\}$, $r_g^t(t = 1, 2, \cdots, n)$ represents the rank of feature t for g-th feature selector. Noting that the ranking value for a feature is set as follows: The best feature with the largest weight is assigned rank 1, and the worst one rank n. Then all the l feature ranking results are aggregated into a consensus feature ranking r_f by linear aggregation, whose results in a sum where features contribute in a linear way with respect to their ranks.

$$r_f^t = \sum_{g=1}^{l} r_g^t \tag{10}$$

4.2 Stability Estimation

To measure the effect of instance perturbation on the feature selection stability, we also adopt a subsampling based strategy. Consider the data set S with N

instances and n features. Then z subsamples of size $\alpha N(0 < \alpha < 1)$ are drawn randomly from S, where the parameters z and α can also be varied. Subsequently, feature selection(both the single and ensemble) is performed on each of the z subsamples, and a measure of stability or robustness is calculated. Here, feature stability is measured by comparing similarity of the outputs of the feature selectors on the z subsamples. The more similar all outputs are, the higher the stability measure will be. The overall stability can be defined as the average similarity over all pairwise similarity comparisons between the different feature selectors:

$$R_{sta} = \frac{1}{z(z-1)} \sum_{u=1}^{z-1} \sum_{v=u+1}^{z} Sim(r_u, r_v) \tag{11}$$

where $Sim(r_u, r_v)$ represents a similarity measure between feature selection results r_u and r_v. For feature ranking, the Spearman rank correlation coefficient [6] can be used to calculate the similarity:

$$Sim(r_u, r_v) = 1 - 6 \sum_{t=1}^{n} \frac{(r_u^t - r_v^t)^2}{n(n^2-1)} \tag{12}$$

5 Experiments

First, experiments are conducted on both real-world and synthetic data sets to see whether the proposed method can rank the important features at the top ranking position. Second, we empirically evaluate the performance of L-Lmba, Lmba [9] and Relief [10,11] , on several real-world data sets. Finally, we evaluate the stability of ensemble feature selection on high dimensional data sets with small samples.

The real-world and synthetic data sets are used to check the correctness of L-Lmba are introduced in Table 1. The ranking results of L-Lmba are also shown in the last column (from left to right) of Table 1. Other real-world benchmark data sets without any prior knowledge are used to test the classification performance of L-Lmba, Lmba and Relief, and the high-dimensional data sets with small samples (Arcene and Colon) are used to validate the stability of ensemble L-Lmba. These data sets are introduced in Table 2, which are taken from UCI ML repository [12] and Colon cancer diagnosis data set is introduced in [13].

5.1 Experimental Results for Single Feature Selection

For data sets with more than two classes, we use the accuracy rate of 1-NN classifier to evaluate the selected feature subsets. For 2-class data sets, linear

Table 1. Ranking results on benchmark and synthesis data sets

Data sets	♯ features	♯ classes	SN of important features	Ranking results
Iris	4	3	3,4	{3,4},1,2
Monk3	6	2	2,4,5	{5,4,2}, 1, 3, 6
Multi-class	10	4	1,2	{2,1},6,5,....

Table 2. Summary of benchmark data sets

Data sets	♯ features	♯ instances	♯ classes
Arcene	10000	200	2
Parkinsons	23	197	2
Multi-features	649	2000	10
Colon	2000	62	2

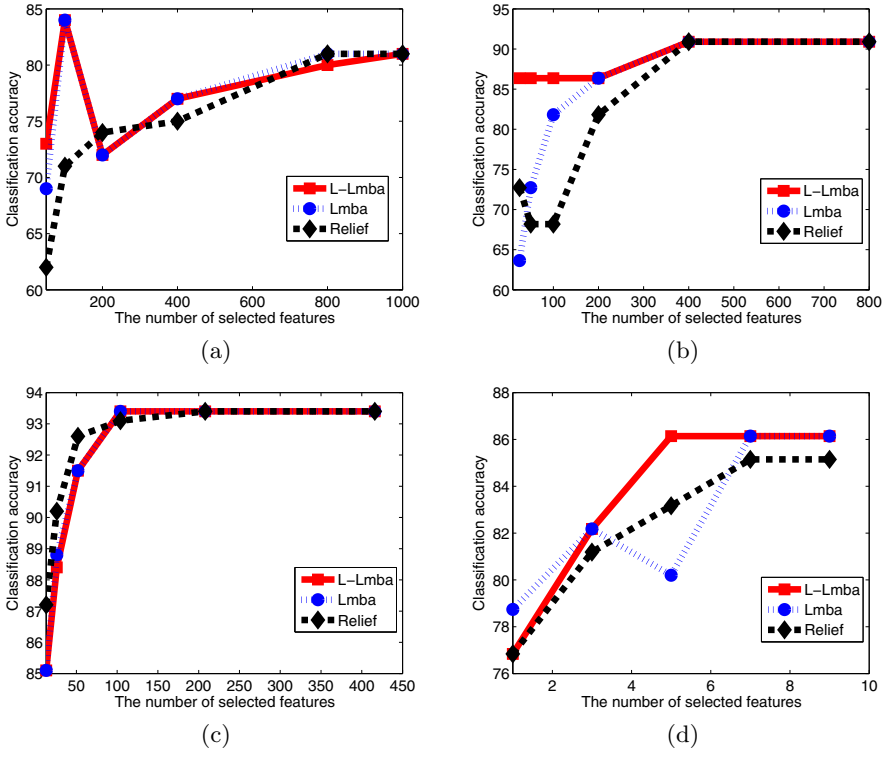


Fig. 1. Experimental results for (a)Arcene; (b)Colon; (c)Multi-features (d)Parkinson

Support Vector Machines (SVM)[14] is adopted, and the parameter C is set to 1. 5-fold cross-validation is adopted. In L-Lmba and Lmba, the parameter c and the number of target neighbors k are set to 1 and 3 respectively.The experimental results are shown in Figure 1(a), (b), (c) and (d), which are corresponding to Arcene, Colon, Multi-features and Parkinson respectively. The X-axis is the number of selected features, and Y-axis is the corresponding accuracy rate.

From the experimental results, we can observe that L-Lmba often obtain higher or similar performance to Relief and Lmba in most cases.

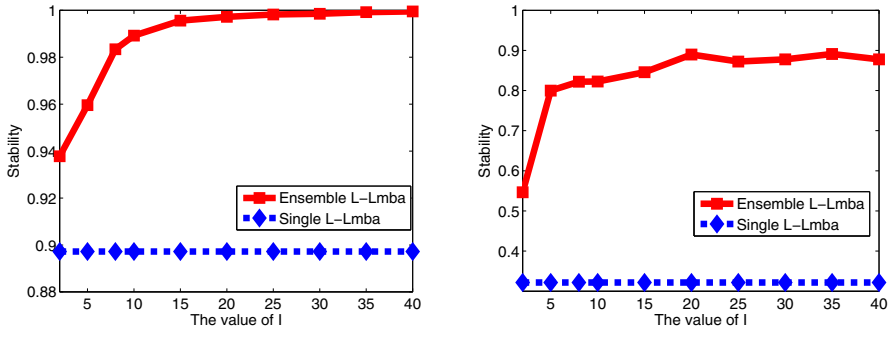


Fig. 2. Experimental results of stability for (a)Arcene; (b)Colon

5.2 Experimental Results for Ensemble Feature Selection

To estimate the robustness (stability) of feature selection algorithm, the strategy explained in section 4.2 was used with $z = 10$ subsamples of size 0.9N (i.e. $\alpha = 0.9$ and each subsample contains 90% of the data). This percentage was chosen because we want to assess robustness with respect to relatively small changes in the data set. Then, ensemble version of L-Lmba was run on each subsample, the features are been ranking based on their weights, and then the similarity between feature ranking results pairs and stability is calculated using (12) and (11) respectively. The effect of the number of feature selectors, i.e. the value of l, on the robustness of the ensemble feature selection is shown in Figure 2 for Arcene and Colon. In general, stability is mostly increased in the first steps, and slows down after about 20 selectors in the ensemble. The stability of single L-Lmba is 0.8936 and 0.3163 for Arcene and Colon respectively, then ensemble feature selection algorithm has higher stability than single one.

6 Conclusions

In the paper, an unified energy-based framework for feature ranking is proposed, and an empirical feature selection algorithm-L-Lmba stems from the framework is also introduced and analyzed. The experimental results have shown that the high performance of our works on benchmark data sets and Colon data. Of course, other feature ranking algorithms can be derived from the proposed framework: Many other loss functions(square-exponential loss, etc) and energy functions(reconstruct error, etc) can be utilized to obtain different evaluation functions for feature ranking. On the other hand, the ensemble version of our proposed algorithm is introduced and validated to show that it can largely improve the stability of feature selection.

Acknowledgments. This research was partially supported by the National Natural Science Foundation of China via the grant NSFC 61073114, Natural

Science Fund for Colleges and Universities in Jiangsu Province via the Grants 09KJB510012 and Scientific Research Foundation of Nanjing University of Posts and Telecommunications (NY210010,NY209003).

References

1. Liu, H., Yu, L.: Toward Integrating Feature Selection Algorithms for Classification and Clustering. IEEE Trans. Knowledge and Data Engineering 17, 494–502 (2005)
2. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 31, 157–1182 (2003)
3. LeCun, Y., Chopra, S., Hadsell, R., Huang, F.J., Ranzato, M.A.: A Tutorial on Energy-Based Learning. In: Bakir, et al. (eds.) Predicting Structured Outputs. MIT Press (2006)
4. Han, Y., Yu, L.: A Variance Reduction Framework for Stable Feature Selection. In: Proc. Int'l Conf. Data Mining, pp. 206–215 (2010)
5. Loscalzo, S., Yu, L., Ding, C.: Consensus Group Based Stable Feature Selection. In: Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2009), pp. 567–576 (2009)
6. Saeys, Y., Abeel, T., Van de Peer, Y.: Robust Feature Selection Using Ensemble Feature Selection Techniques. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 313–325. Springer, Heidelberg (2008)
7. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance Metric Learning for Large Margin Nearest Neighbor Classification. In: Advances in Neural Information Processing Systems, Cambridge MA, vol. 18 (2006)
8. Weinberger, K.Q., Saul, L.K.: Distance Metric Learning for Large Margin Nearest Neighbor Classification. Journal of Machine Learning Research 10, 207–214 (2009)
9. Li, Y., Lu, B.L.: Feature Selection Based on Loss Margin of Nearest Neighbor Classification. Pattern Recognition 42, 1914–1921 (2009)
10. Kira, K., Rendell, L.: A Practical Approach to Feature Selection. In: Proc. of Int'l Conf. on Machine Learning, pp. 249–256 (1992)
11. Robnik-Sikonja, M., Kononerko, I.: Theoretical and Empirical Analysis of Relief and ReliefF. Machine Learning, 23-69 (2003)
12. Merz, C.J., Murphy, P.M.: UCI repository of machine learning database (1996), http://www.ics.uci.edu/mlearn/MLRepository.html
13. Alon, U., Barkai, N., et al.: Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Cancer Tissues Probed by Oligonucleotide Arrays. Proc. of the National Academy of Sciences of the United States of America 96, 6745–6750 (1999)
14. Chang, C.C., Lin, C.J.: LIBSVM: a Library for Support Vector Machines, http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.ps.gz

The Rough Set-Based Algorithm for Two Steps

Shu-Hsien Liao[1], Yin-Ju Chen[2], and Shiu-Hwei Ho[3]

[1] Department of Management Sciences, Tamkang University,
No.151 Yingzhuan Rd., Danshui Dist., New Taipei City 25137, Taiwan R.O.C
[2] Graduate Institute of Management Sciences, Tamkang University,
No.151 Yingzhuan Rd., Danshui Dist., New Taipei City 25137, Taiwan R.O.C
[3] Department of Business Administration, Technology and Science Institute of Northern Taiwan,
No. 2, Xueyuan Rd., Peitou, 112 Taipei, Taiwan, R.O.C
michael@mail.tku.edu.tw, s5515124@ms18.hinet.net,
succ04.dba@msa.hinet.net

Abstract. The previous research in mining association rules pays no attention to finding rules from imprecise data, and the traditional data mining cannot solve the multi-policy-making problem. Furthermore, in this research, we incorporate association rules with rough sets and promote a new point of view in applications. The new approach can be applied for finding association rules, which has the ability to handle uncertainty combined with rough set theory. In the research, first, we provide new algorithms modified from Apriori algorithm and then give an illustrative example. Finally, give some suggestion based on knowledge management as a reference for future research.

Keywords: Rough sets, Association rules, Data mining, Knowledge management.

1 Introduction

In data mining, it is often unclear which algorithm is best suited for the problem. Here, we require some decision support for data mining. To ensure that appropriate data is recorded when the collection process begins, it is useful to first build a decision model and use it as a basis for defining the attributes that will describe the data [2]. First, the quick growth of the Internet has led the global economy and social reforming. The traditional business model has been changed; the network shopping has become a part of our daily lives. Therefore, the fast growth of Internet stores changes not only the way customers buy goods but also the way customers receive goods. This means that the customers will pay more and more attention to the logistics services of the Internet stores, and then shopping on the Internet has become an important marketing channel. Second, most customers will rely on the recommendation system for a web-based computer retail store. It is a systematic attempt to strengthen a firms' competitive ability or give useful information to the end user. For the trend of diverseness, we propose a mining concept for knowledge refinement that is based on the discovery of unexpected patterns and uncertain information from the multi-database and the result of accurate prediction can be used for recommending products. The remainder of this paper is organized as follows. Section 2 reviews relevant literature and the problem statement. Section 3 new algorithms modified from Apriori algorithm. Section 4 an illustrative example. Closing remarks and future work are presented in Section 5.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 63–70, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Literature Review and Problem Statement

Rough set theory, proposed by Pawlak in the 1980s, is a theory for the study of intelligent systems characterized by inexact, uncertain or vague information. Now rough set theory has found successful applications in such fields of artificial intelligence as machine learning, knowledge discovery, decision analysis, process control, pattern recognition, etc. It has become one of flash points in the research area of information science [3, 10]. An information system is a 4-tuple S={U, A, V, f}, where U is a finite set of objects, called the universe, A is a finite set of attributes, V is a domain of attribute a, and f:U×A→V is called an information function such that f(x,a) [5]. RST is an approach to aid decision making in the presence of uncertainty. It classifies imprecise, uncertain or incomplete information expressed in terms of data acquired from experience. In RST, a set of all similar objects is called an elementary set, which makes a fundamental atom of knowledge. Any union of elementary sets is called a crisp set and other sets are referred to as rough set. As a result of this definition, each rough set has boundary-line elements. For example, some elements cannot be definitively classified as members of the set or its complement. In other words, when the available knowledge is employed, boundary-line cases cannot be properly classified. Therefore, rough sets can be considered as uncertain or imprecise [7]. The association rule has become one of the most important techniques in data mining. The Apriori algorithm is a great discovery for data mining and the association rule. By reducing insignificant candidate item sets, it successfully increases processing speed and reduces the usage of space [1]. Traditional data mining cannot solve the multi-policy-making problem, such as, the consumer buys two things at the same time. In the real world, if the customers transaction data is either A or B, it does not differ to transform categorical or nominal data into Boolean. If we examine from Table 1, Table 2 does not look different. However, if the customers transaction dates are A and B, and if we examine from Table 3, Table 4 will look different. It will be the key factors for mining consumer purchasing patterns in transaction databases.

Table 1. Original data (Either A or B)

U \ Q	Gender	Age	Purchasing
001	Male	20	Coca-Cola
002	Female	23	Pepsi
003	Male	30	Coca-Cola
004	Male	22	Pepsi

Table 2. Original data (Boolean type)

U \ Q	Gender		Age			Purchasing	
	Male	Female	Under 20	20~25	26~30	Coca-Cola	Pepsi
001	1	0	0	1	0	1	0
002	0	1	0	1	0	0	1
003	1	0	0	0	1	1	0
004	1	0	0	1	0	0	1

Table 3. Original data (Both A and B)

Q / U	Gender	Age	Purchasing
001	Male	20	Coca-Cola & Pepsi
002	Female	23	Pepsi
003	Male	30	Coca-Cola
004	Male	22	Coca-Cola & Pepsi

Table 4. Original data (Boolean type)

Q	Gender		Age			Purchasing	
U	Male	Female	Under 20	20–25	26–30	Coca-Cola	Pepsi
001	1	0	0	1	0	1	1
002	0	1	0	1	0	0	1
003	1	0	0	0	1	1	0
004	1	0	0	1	0	1	1

3 Rough Set-Based Algorithm for Two-Step

In the classic rough set theory (RST), the attributes considered in an information system, including decision attributes, are taken on nominal values [4]. Mining generalized association rules between items in the presence of taxonomies has been recognized as an important model in data mining [6]. For the trend of diverseness market, the traditional association rule algorithm concept does not meet the needs of consumers anymore. The RST has been successfully applied in selecting attributes for improving the effectiveness in deriving decision trees/rules for decisions and classification problems [2]. The new algorithm modified from Apriori algorithm that is based on the RST has the ability to handle the uncertainty in the classing process and finding association rule. The new algorithm modified from rough set and Apriori is reproduced below.

Table 5. Sample data set

Q	Attributes			Decision
U	Age	Gender	Shopping frequency	Brand loyalty
t_1	20–29	Male	Once a month	High
t_2	30–39	Female	Under fortnight	Median
t_3	20–29	Male	Once a month	High
t_4	20–29	Female	Once a month	Median
t_5	10–19	Male	Other	Low
t_6	30–39	Female	Under fortnight	Median
t_7	10–19	Male	Other	Low

Step 1: The basic concepts of the RST, an information system, can be seen as a system. We input data to reduce attributes as shown in Table 5. There are ten objects DT = {t_1, t_2,…, t_{10}}, three attributes A = {Age, Gender, Shopping frequency}, and a

decision attribute D = {Brand loyalty}. Assume the decision attribute has only three possible values: {High, Low, Median}, and the other input data is as shown in Table 5.

Step 2: Then we try to find $B = [t_i]_{Ind(A)}$ in Table 5. In the example, we classified customers into three major profitable groups with decision attribute. Therefore, three classes exist in the data set shown in Table 6.

Table 6. Attributes set

Q / U	Attributes			Decision
	Age	Gender	Shopping frequency	Brand loyalty
$\{t_1, t_3\}$	20–29	Male	Once a month	High
$\{t_2, t_6\}$	30–39	Female	Under fortnight	Median
$\{t_4\}$	20–29	Female	Once a month	Median
$\{t_5, t_7\}$	10–19	Male	Other	Low

Table 7. Ind(A)= Ind(A-a_j)(a_j=Age)

Q / U	Attributes		Decision
	Gender	Shopping frequency	Brand loyalty
$\{t_1, t_3\}$	Male	Once a month	High
$\{t_2, t_6\}$	Female	Under fortnightly	Median
$\{t_4\}$	Female	Once a month	Median
$\{t_5, t_7\}$	Male	other	Low

Table 8. Ind(A) ≠ Ind(A-aj)(a_j = Shopping frequency)

Q / U	Attributes		Decision
	Age	Gender	Brand loyalty
$\{t_1, t_3\}$	20–29	Male	High
$\{t_2, t_6\}$	30–39	Female	Median
$\{t_4\}$	20–29	Female	Median
$\{t_5, t_7\}$	10–19	Male	Low

Table 9. Ind(A) ≠ Ind(A-a_j) (a_j = Gender)

Q / U	Attributes		Decision
	Age	Shopping frequency	Brand loyalty
$\{t_1, t_3, t_4\}$	20–29	Once a month	High
$\{t_2, t_6\}$	30–39	Under fortnightly	Median
$\{t_5, t_7\}$	10–19	other	Low

Step 3: According to Table 6, discriminate the efficient attribute set, which is used in mining association rules and building purchase profile. If Ind(A) = Ind(A-a_j) are shown in Table 7, Ind(A) ≠ Ind(A-a_j) are shown in Table 8 and Table 9. Finally, we find Ind(B) = {Gender, Shopping frequency} is the indiscernibility relation (see Table 10). Then, we use the reduce attribute set to find the customer who has high brand loyalty.

Table 10. Final data set

Q / U	Attributes		Decision
	Gender	Shopping frequency	Brand loyalty
$\{t_1, t_3\}$	Male	Once a month	High
$\{t_2, t_6\}$	Female	Under fortnightly	Median
$\{t_4\}$	Female	Once a month	Median
$\{t_5, t_7\}$	Male	other	Low

Step 4: Using the Apriori algorithm and mining association rules by minimum support and minimum confidence.

$$\text{Support}((\text{Male}) \cap (\text{Once a month})) = \frac{(\text{Male}) \cap (\text{Once a month}) \text{Total of trades in database}}{\text{Total of trades in database}} = \frac{1}{5} = 20\%$$

$$\text{Confidence}((\text{Male}) \cap (\text{Once a month}) \rightarrow (\text{Brand loyalty - high})) = \frac{(\text{Male}) \cap \text{Once a month} \cap (\text{Brand loyalty - high}) \text{Total of trades in database}}{(\text{Male}) \cap (\text{Once a month}) \text{Total of trades in database}} = \frac{1}{1} = 100\%$$

Algorithm-First stage

```
Input:
Information System (IS);
Output:
{ Attribute Sets};
Method:
 1. Begin
 2.   IS = (U, A);
 3.    x_1, x_2, ..., x_n ∈ U ; /* where x_1, x_2, ..., x_n are the objects of
       set U */
 4.    a_1, a_2, ..., a_m ∈ A ; /* where a_1, a_2, ..., a_m are the elements of
       set A */
 5.  For each a_m do;
 6.   compute f(t, a); /* compute the information function
          in IS */
 7.   compute Ind(A − a_j); /* compute the relative reduct of
          the elements for element j */
 8.  Endfor;
 9.   Output { Attribute Sets };
10. End;
```

Algorithm-Second stage

```
Input:
Decision Table (DT);
Output:
{Classification Rules};
```

```
Method:
 1. Begin
 2. DT = (U, Q);
 3.     t1, t2, ⋯ ti ∈ U ; /* where t1, t2, ⋯ ti are the objects of
        set U */
 4.     Q = (A, D);
 5.      a1, a2, ⋯, aj ∈ A ; /* where a1, a2, ⋯, aj are the
          elements of set A */
 6.      d1, d2, ⋯, dl ∈ D ; /* where d1, d2, ⋯, dl are the decision
          elements of set D */
 7.  For each dl do;
 8.    compute f(t, a); /* compute the information function
        in DT */
 9.    compute Sup(aj); /* compute the support */
10.    compute Conf(aj → Dk); /* compute the confidence */
11.  Endfor;
12.    Output {Classification Rules};
13.End;
```

4 Illustrative Example

The two steps approach is based on a though market understanding and has to be managed in such a way as to effectively meet differing customer needs. Link up database marketing concept in the first step with the help of segmentation. Then, Apriori algorithm provides us with this possibility to categorize customers with most similarities in a group. In the present competitive environment within the service industries, retailing customers a maximizing profit from an existing customer base has become at least as important as attracting new customers. The input variable for each customer includes "education," "age," "shopping frequency," "product using state," "e-paper," et al. According to those attributes, we find some rules for decision criteria. We arrange the rank of decision rules (Table 11 and Table 12).

In first step, the rough set is applied to reduce attributes, and based on those attributes, we find some rules for the decision criteria. In the second step, the association rule is help to find an association diagram from customer's transaction data (Table 13 and Table 14). The two-step approach is easier than traditional method in finding association rule.

Table 11. Decision rules (output: price sensitive)

No	Attribute set	rules	Generate rule
1	Education, product using state	227	(education) & (product using state) => (price sensitive)
2	Age, product using state	191	(age) & (product using state) => (price sensitive)

Table 12. Decision rules (output: brand loyalty)

No	Attribute set	rules	Generate rule
1	Education, product using state, shopping frequency	227	(education) & (product using state) & (shopping frequency) => (brand loyalty)
2	Education, product using state	203	(education) & (product using state) => (brand loyalty)
3	Education, e-paper	185	(education) & (e-paper) => (brand loyalty)

Table 13. Association rule (min Sup=10, min Con=80)

Sup	Conf	Lift	Consequent	Antecedent		
15.5	92.5	1.2	Brand loyalty (Medium)	Price Sensitivity (High)	University or college degree	e-paper (No)
12.3	88.1	1.2	Brand loyalty (Medium)	Price Sensitivity (Medium)	University or college degree	e-paper (No)
11.4	87.2	1.1	Brand loyalty (Medium)	aged 20–29	Family use	e-paper (No)

Table 14. Association rule (min Sup = 5, min Con = 50)

Sup	Conf	Lift	Consequent	Antecedent			
6.5	59.1	2.6	Price Sensitivity (Low)	Brand loyalty (Low)	University or college degree	e-paper (No)	-
6.2	52.4	1.5	Price Sensitivity (High)	Personal use	University or college degree good	Brand loyalty (Medium)	e-paper (No)
7.0	50.0	1.6	Price Sensitivity (Medium)	Family use	Brand loyalty (Medium)	e-paper (No)	aged 50–59

5 Conclusion

Rough Set Association Rule (RSAR) algorithm that is based on the RST has the ability to handle the uncertainty in the classing process and finding the association rule. Using the suggested methodology can reduce the dimension of transactions and help the decision maker to find useful association rules more effective than tradition association rules. Because using the traditional association rule to calculate support and confidence, as we changed min support and min confidence once, it will calculate again. In the example, we have to calculate $C_1^3 \times C_1^2 \times C_1^4 \times C_1^3 = 72$. In the new algorithm given in the example, we only need to calculate $C_1^2 \times C_1^4 \times C_1^3 = 24$. The method in the paper, we can use them to calculate itemsets' support and confidence from imprecise and multiple criteria data.

In this paper, we focus on the concept of rule and the management of customer relationship in the market downturn. Using the suggested methodology, decision maker can accurate at segmentation and assist the development of a new product. The

result of accurate prediction also can be used for recommending products to the customers and suggesting useful links. The strategic challenges faced by organizations are often framed in information- and knowledge-based terms such as uncertainty or complexity [9]. Consequently, how to find appropriate solutions to their particular knowledge problems is a most important problem for organizations to implement KM and make decisions. Real-world data tends to be imprecise due to human errors, instrument errors, recording errors, and so on. The RSAR algorithm has the ability to handle uncertainty and missing data in the classing process and discovering meaningful patterns and rules. This study proposes a new algorithm that can transform knowledge into information. Meanwhile, the knowledge conversion process to resolve the complexity that exists and provide more accurate information to make decisions of policy makers. Mining activities help us to know data patterns.

Acknowledgements. This research was funded by the National Science Council, Taiwan, Republic of China, under contract NSC 100-2410-H-032 -018-MY3.

References

1. Chen, C.M., Liao, S.H.: Association Rule Algorithms for Logical Equality Relationships. In: IEEE 8th International Conference on Computer and Information Technology, Sydney, Australia, July 8-11 (2008)
2. Lavrac, N., Bohanec, M., Pur, A., Cestnik, B., Debeljak, M., Kobler, A.: Data mining and visualization for decision support and modeling of public health-care resources. Journal of Biomedical Informatics 40, 438–447 (2007)
3. Lee, J.W.T., Yeung, D.S., Tsang, E.C.C.: Soft Computing - A Fusion of Foundations, Methodologies and Applications. Soft Computer 49(1), 27–33
4. Lee, J.W.T., Yeung, D.S., Tsang, E.C.C.: Rough sets and ordinal reducts. Soft Computing - A Fusion of Foundations, Methodologies and Applications 10, 27–33 (2006)
5. Li, R., Wang, Z.-o.: Mining classification rules using rough sets and neural networks. European Journal of Operational Research 157(2), 439–448 (2004)
6. Lian, W., Cheung, D., Yiu, S.M.: An efficient algorithm for finding dense regions for mining quantitative association rules. Computers and Mathematics with Applications 50(3-4), 471–490 (2005)
7. Parmar, D., Wu, T., Blackhurst, J.: MMR: An algorithm for clustering categorical data using. Rough Set Theory, Data & Knowledge Engineering 63(3), 879–893 (2007)
8. Uta, J., Martin, C., Susan, B.: Demand chain management-integrating marketing and supply chain management. Industrial Marketing Management 36(3), 377–392 (2007)
9. Zack, M.H.: The role of decision support systems in an indeterminate world. Decision Support Systems 43(4), 1664–1674 (2007)
10. Zhang, W.X., Qiu, G.F., Wu, W.Z.: A general approach to attribute reduction in rough set theory. Science in China Series F: Information Sciences 50(2), 188–197 (2007)

An Infinite Mixture of Inverted Dirichlet Distributions

Taoufik Bdiri and Nizar Bouguila

Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada, Qc, H3G 2W1
t_bdiri@encs.concordia.ca, bouguila@ciise.concordia.ca

Abstract. In this paper we present an infinite mixture model based on inverted Dirichlet distributions. The proposed mixture is learned using a fully Bayesian approach and allows to overcome a challenging issue when dealing with data clustering namely the automatic selection of the number of clusters. We explore the performance of the proposed approach on the challenging problem of text categorization. The results show that the proposed approach is effective for positive data modeling when compared to those reported using infinite Gaussian mixture.

Keywords: Inverted Dirichlet, Dirichlet process, model selection.

1 Introduction

Recently we have seen a growing interest in methods for learning from data. The main goal has been to extract useful knowledge hidden in these data. Among the many employed data-driven learning techniques, complex mixture models have been widely used naturally in several situations and applications [1,2,3]. In particular, extensive works have been devoted to the Gaussian mixture (see, for instance, [4,5]). However, the standard assumption that the data is Gaussian is generally inappropriate and far from satisfactory [6]. Indeed, the choice of an appropriate learning technique should depend on the nature of the tackled data and the form of available knowledge. This is especially true in the case of positive data as we have shown in [7] where a finite mixture model based on inverted Dirichlet distributions has been proposed. The model in [7] has been learned (i.e. parameters estimation and model selection) using a minimum message length objective function by considering a finite number of mixture components. Novel approaches, however, have been based on the consideration of infinite models which have proven to be a powerful alternative for model selection in a very wide variety of situations and applications. The goal of this paper is to extend our work in [7] to the infinite case by considering a nonparametric Bayesian approach namely Dirichlet processes which have a history going back to Antoniak [8]. With infinite mixture models, we suppose actually that vectors are assigned as they arrive to existing clusters or to a new created one which provides an elegant solution to the clustering problem by eliminating the need to specify a number of mixture components a priori.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 71–78, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Dirichlet processes and its many generalizations have seen a renewed interest in the fields of neural networks, pattern recognition and computer vision, in part due to the arrival of efficient MCMC techniques to perform sampling [9,10,11,12,13,14]. Moreover, as a nonparametric Bayesian formalism, infinite models allow the representation of uncertainties in both model's parameters and structure through the incorporation of prior knowledge and the combination of this knowledge with the data which results in posteriors [14]. In our work we shall use two MCMC techniques to infer the cluster structure namely Gibbs sampling, popularized in [15] and summarized in [16], and Metropolis-Hastings algorithm [17].

The rest of this paper is organized as follows. First and immediately below our infinite inverted Dirichlet model is presented and an approach to learn it is proposed. Section 3, presents the experiments and discusses the obtained results. Finally, in section 4, we draw our conclusions.

2 The Infinite Model

We describe here our infinite model, how it is learnt, by specifying its parameters priors and posteriors, and we give the complete learning algorithm.

2.1 The Finite Inverted Dirichlet Mixture Model

If a D-dimensional positive vector $\boldsymbol{X} = (X_1, X_2, \ldots, X_D)$ follows an inverted Dirichlet distribution, the joint density function is given by [18]

$$p(\boldsymbol{X}|\boldsymbol{\alpha}) = \frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{d=1}^{D+1}\Gamma(\alpha_d)} \prod_{d=1}^{D} X_d^{\alpha_d - 1}(1 + \sum_{d=1}^{D} X_d)^{-|\boldsymbol{\alpha}|} \tag{1}$$

where $X_d > 0, d = 1, 2, \ldots, D$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{D+1})$ is the vector of parameters and $|\boldsymbol{\alpha}| = \sum_{d=1}^{D+1} \alpha_d$, $\alpha_d > 0, d = 1, 2, \ldots, D+1$. The inverted Dirichlet has a number of desirable properties such as its flexibility and ease of use. Indeed, it allows multiple symmetric and asymmetric modes since it may be skewed to the right, skewed to the left or symmetric [7]. By substituting $\mu_d = \frac{\alpha_d}{|\boldsymbol{\alpha}|}$ in the previous equation, the inverted Dirichlet distribution can be written as following

$$p(\boldsymbol{X}|\boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{d=1}^{D+1}\Gamma(\mu_d|\boldsymbol{\alpha}|)} \prod_{d=1}^{D} X_d^{\mu_d|\boldsymbol{\alpha}|-1}(1 + \sum_{d=1}^{D} X_d)^{-|\boldsymbol{\alpha}|} \tag{2}$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{D+1})$. We shall adopt this parametrization, since it will allow the specification of accurate priors over our model's parameters as we shall see in next section. An inverted Dirichlet mixture with M components is defined as

$$p(\boldsymbol{X}|\xi) = \sum_{j=1}^{M} p(\boldsymbol{X}||\boldsymbol{\alpha}_j|, \boldsymbol{\mu}_j)P_j \tag{3}$$

where P_j ($0 < P_j < 1$ and $\sum_{j=1}^{M} P_j = 1$) are the mixing parameters and $p(\boldsymbol{X}||\boldsymbol{\alpha}_j|, \boldsymbol{\mu}_j)$ is the inverted Dirichlet distribution. The symbol ξ refers to the entire set of parameters to be estimated $\xi = (\{\xi_j\}, \boldsymbol{P})$, where $\xi_j = (|\boldsymbol{\alpha}_j|, |\boldsymbol{\mu}_j|)$ and $\boldsymbol{P} = (P_1, \ldots, P_M)$.

2.2 The Infinite Inverted Dirichlet Mixture Model

Let $\mathcal{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N)$ be a set of independent positive vectors representing N objects (e.g. image, text, video, etc.). In the following, we shall assume that $\mathcal{X}$ is modeled by a Dirichlet process of inverted Dirichlet distributions. A Dirichlet process mixture model of inverted Dirichlet distributions is a mixture model where the number of components is supposed to be infinite. It is different from a finite mixture of inverted Dirichlet distributions model where each vector $\boldsymbol{X}_i$ is supposed to be drawn from one of M inverted Dirichlet distributions with different parameters $\xi_1, \ldots, \xi_M$. Indeed, in the infinite case, we assume that the number of clusters may increase as new data arrive by supposing that $\boldsymbol{X}_i$ follows a model where the parameter ξ are generated from a distribution G as follows:

$$\begin{aligned} \xi_i|G &\sim G, i = 1, \ldots, N \\ G &\sim DP(G_0, \eta) \\ \boldsymbol{X}_i|\xi_i &\sim p(\boldsymbol{X}_i|\xi_i), i = 1, \ldots, N \end{aligned} \tag{4}$$

where $p(\boldsymbol{X}_i|\xi_i)$ is an inverted Dirichlet distribution that $\boldsymbol{X}_i$ follows, and G_0 and η define a baseline distribution for the Dirichlet process prior and the concentration parameter, respectively. The consideration of Dirichlet process in clustering allows to handle implicitly the problem of how to determine the number of clusters. Indeed, it can be viewed as the limit of a finite mixture model with a uniform Dirichlet prior (i.e. a Dirichlet with hyperparameters η/M, where M is the number of clusters) [4].

We let $Z = (Z_1, \ldots, Z_N)$ denote the membership vectors of the data to cluster in $\mathcal{X}$, where each Z_i is an integer denoting the unknown component from which $\boldsymbol{X}_i$ is drawn. Under the infinite mixture model it is possible to show that [4]:

$$p(Z_i = j|Z_1, \ldots, Z_{N-1}) = \begin{cases} \frac{n_j}{N-1+\eta} \text{ if } n_j > 0 \text{ (cluster } j \in \mathcal{R}) \\ \frac{\eta}{N-1+\eta} \text{ if } n_j = 0 \text{ (cluster } j \in \mathcal{U}) \end{cases} \tag{5}$$

where n_j is the number of vectors previously affected to cluster j, and $\mathcal{R}$ and $\mathcal{U}$ are the sets of represented and unrepresented clusters, respectively. An important advantage here is that we allow uncertainty in the number of mixture components and for clustering [1]. Indeed, the number of clusters during sampling could increase by one, decrease by one, or remain unchanged which avoids the need to assume a fixed number of classes as in the case of finite mixture models.

[1] The approach can be viewed actually as a Bayesian way to average over different models. It is noteworthy that some deterministic model averaging approaches have bee proposed, also (see, for instance, [19]).

The Dirichlet process mixture approach for clustering is based on the MCMC technique of Gibbs sampling [20] by generating the assignments of vectors according to the posterior distribution

$$p(Z_i = j|Z_{-i}, \mathcal{X}) \propto p(Z_i = j|Z_{-i}) \int p(\boldsymbol{X}_i|Z_i = j, \xi_j)p(\xi_j|Z_{-i}, \mathcal{X}_{-i})d\xi_j \quad (6)$$

where Z_{-i} represents all the vectors assignments except Z_i and $\mathcal{X}_{-i}$ represents all the vectors except $\boldsymbol{X}_i$.

2.3 Priors and Conditional Posteriors

The main goal here is to obtain the conditional posterior distributions of our infinite model's parameters given the data to cluster. This requests the choice of prior distributions. In this paper, we impose independent uniform and inverse Gamma priors on the $\boldsymbol{\mu}_j$ and $|\boldsymbol{\alpha}_j|$ parameters, respectively:

$$p(\boldsymbol{\mu}_j) \sim \prod_{l=1}^{d+1} \mathcal{U}_{[0,1]}^{jl} \qquad p(|\boldsymbol{\alpha}_j||\sigma, \varpi) \sim \frac{\varpi^\sigma \exp(-\varpi/|\boldsymbol{\alpha}_j|)}{\Gamma(\sigma)|\boldsymbol{\alpha}_j|^{\sigma+1}} \quad (7)$$

where σ and ϖ are hyperparameters, common to all components, representing shape and scale of the distribution, respectively. Having these priors, the conditional posteriors for $|\boldsymbol{\alpha}_j|$ and $\boldsymbol{\mu}_j$ given the rest of the variables are given by:

$$p(|\boldsymbol{\alpha}_j||\ldots) \propto \frac{\varpi^\sigma \exp(-\varpi/|\boldsymbol{\alpha}_j|)}{\Gamma(\sigma)|\boldsymbol{\alpha}_j|^{\sigma+1}} \left[\frac{\Gamma(|\boldsymbol{\alpha}_j|)}{\prod_{d=1}^{D+1}\Gamma(\mu_{jd}|\boldsymbol{\alpha}_j|)}\right]^{n_j} \times \prod_{Z_i=j}\left[\prod_{d=1}^{D} X_{id}^{\mu_{jd}|\boldsymbol{\alpha}_j|-1}(1+\sum_{d=1}^{D} X_{id})^{-|\boldsymbol{\alpha}_j|}\right] \quad (8)$$

$$p(\boldsymbol{\mu}_j|\ldots) \propto \left[\frac{\Gamma(|\boldsymbol{\alpha}_j|)}{\prod_{d=1}^{D+1}\Gamma(\mu_{jd}|\boldsymbol{\alpha}_j|)}\right]^{n_j} \prod_{Z_i=j}\left[\prod_{d=1}^{D} X_{id}^{\mu_{jd}|\boldsymbol{\alpha}_j|-1}(1+\sum_{d=1}^{D} X_{id})^{-|\boldsymbol{\alpha}_j|}\right] \quad (9)$$

where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_i=j}$ and represents the number of vectors belonging to cluster j. For greater flexibility of our model, the hyperparameters, σ and ϖ, associated with the $|\boldsymbol{\alpha}_j|$ shall be also considered as random variables and then given inverse Gamma and exponential priors, respectively:

$$p(\sigma|\lambda, \delta) \sim \frac{\delta^\lambda \exp(-\delta/\sigma)}{\Gamma(\lambda)\sigma^{\lambda+1}} \qquad p(\varpi|\phi) \sim \phi \exp(-\phi\varpi) \quad (10)$$

And their posteriors can be easily obtained using Eqs. 7 and 10:

$$p(\sigma|\ldots) \propto \frac{\varpi^{M\sigma}\delta^\lambda \exp(-\delta/\sigma)}{\Gamma(\sigma)^M\Gamma(\lambda)\sigma^{\lambda+1}} \prod_{j=1}^{M} \frac{\exp(-\varpi/|\boldsymbol{\alpha}_j|)}{|\boldsymbol{\alpha}_j|^{\sigma+1}} \quad (11)$$

$$p(\varpi|\ldots) \propto \frac{\varpi^{M\sigma}\phi\exp(-\phi\varpi)}{\Gamma(\sigma)^M}\prod_{j=1}^{M}\frac{\exp(-\varpi/|\boldsymbol{\alpha}_j|)}{|\boldsymbol{\alpha}_j|^{\sigma+1}} \tag{12}$$

MCMC techniques are very popular in the statistics literature as an approach for sampling from complicated distributions which is actually our case. Thus, we shall use two MCMC techniques for posteriors computation namely Gibbs sampling and Metropolis-Hastings (M-H), and the complete learning algorithm can be summarized as follows:

- Generate $\boldsymbol{Z}_i$ from Eq. 6, $i = 1, \ldots, N$ using the algorithm in [20].
- Update the number of represented components M.
- Update n_j and $P_j = \frac{n_j}{N+\eta}$, $j = 1, \ldots, M$.
- Update the mixing parameters of unrepresented components $P_U = \frac{\eta}{\eta+N}$.
- Generate $\boldsymbol{m}_j$ from Eq. 9 and $\boldsymbol{s}_j$ from Eq. 8, $j = 1, \ldots, M$ using M-H.
- Update the hyperparameters: Generate σ from Eq. 11 and ϖ from Eq. 12 using adaptive rejection sampling as proposed in [21].

3 Experimental Results

To assess the performance of our infinite model, we apply it for the challenging problem of text documents classification. It is important to mention that the main goal of our experiments is to compare the performance of infinite inverted Dirichlet and infinite Gaussian mixture [4] models. The comparison with the many approaches that have been proposed in the past for the problem of text categorization is clearly beyond the scope of this paper. In the following experiments our default choice of the hyperparameters is $(\lambda, \delta, \phi, \eta) = (1, 1, 1/8, 1)$ and the obtained results are averaged over 20 runs of our learning algorithm where each run is based on 20000 samples generated from our Gibbs sampler. It is noteworthy that each time the first 5000 samples are discarded as burn-in.

The main goal of text categorization is to assign automatically documents to a predefined number of semantic categories. Thus, it is crucial to have accurate models that represent effectively the different categories. In our experiments, we describe documents using the so called "bag-of-words" representation where each distinct word in the set of documents corresponds to a feature with its TFIDF score as the value. The evaluation is done on two text classification tasks namely Reuters and WebKB4 [2] data set, which is a subset of the WebKB data set, containing 4199 web pages gathered from university computer science departments and limited to the four most common categories: Course, Faculty, Project, and Student. Table 1 specifies the size of the four categories. For Reuters-21578 and following previous studies that have considered this data set (see, for instance, [22]), we use the "ModApte" subset which contains 12,902 documents grouped into 135 topics from which we consider only the 10 most frequent categories

[2] This data set is available on the Internet. See http://www.cs.cmu.edu/ ∼textlearning

Table 1. Details of the WebKB4 data set

Category	Number of articles	Proportion
Course	930	22.1
Faculty	1124	26.8
Project	504	12.0
Student	1641	39.1

Table 2. Average (± standard deviation) categorization performance (in %) when using infinite inverted Dirichlet mixture for the "ModApte" data set

	Error	Recall	Precision	F_1
earn	5.31± 0.11	88.14±0.42	91.17±0.18	89.85±0.29
acq	6.71±0.22	84.94±0.92	87.90±0.63	85.89±0.74
money-fx	5.19±0.26	58.97±1.77	78.56±1.53	69.96±1.71
grain	3.12±0.18	70.68±2.8	92.80±1.12	80.07±1.95
crude	3.18±0.11	68.80±2.1	85.93±2.23	77.02±1.78
trade	4.01±0.30	56.95±4.10	79.98±2.1	77.22±1.50
interest	4.61±0.39	51.11±4.12	76.86±2.22	60.55±1.98
ship	2.77±0.31	43.40±4.08	88.73±2.12	59.97±1.65
wheat	2.17±0.13	59.94±1.32	91.08±1.42	71.16±1.57
corn	1.89±0.11	48.76±3.13	93.38±1.85	65.54±2.78

Table 3. Average (± standard deviation) categorization performance (in %) when using infinite Gaussian mixture for the "ModApte" data set

	Error	Recall	Precision	F_1
earn	10.13± 0.29	85.21±0.61	88.48±0.24	85.38±0.40
acq	9.34±0.37	80.72±0.78	85.79±0.56	81.73±0.87
money-fx	10.24±0.33	55.87±1.82	75.31±2.01	65.66±1.72
grain	6.74±0.35	67.37±2.72	89.52±1.26	78.01±1.74
crude	6.52±0.39	64.57±2.66	81.63±2.16	74.80±1.59
trade	7.60±0.51	53.70±2.29	76.44±1.91	74.58±1.26
interest	6.95±0.58	49.30±3.21	73.22±2.12	56.87±1.64
ship	4.95±0.42	40.02±3.52	84.58±1.95	55.76±1.38
wheat	4.61±0.13	55.83±1.20	88.37±1.61	67.70±1.22
corn	3.34±0.17	44.72±2.95	90.02±1.95	61.05±2.31

while keeping all their documents. Due to space limitation, we give more details about the results obtained with the "ModApte" data set.

For the "ModApte" subset, we consider training and testing sets of equal size. The vectors representing the different documents in the training set are then modeled by infinite inverted Dirichlet mixtures, using the algorithm in the previous section, and each test vector is affected to the class that gives the highest likelihood. The categorization results are measured in terms of error rate. We also consider other popular measures namely precision, recall and a

well-known combination of them which is the F_1 measure [23]. Tables 2 and 3 give details on the results using both infinite inverted Dirichlet and infinite Gaussian mixture models, respectively. According to these tables, it is clear that the inverted Dirichlet mixture outperforms the Gaussian mixture. For the WebKB4 data set, we first select the top 300 words to be considered for the bag-of-words representation. Moreover, we use different training and testing sets (200 vectors from each category are taken as training sets in each simulation). The average accuracy when using infinite inverted Dirichlet and Gaussian mixtures were 82.59 ± 0.41 and 79.08 ± 0.51, respectively.

4 Conclusion

This paper draws together two flexible models namely inverted distributions and Dirichlet processes which resulted in a flexible infinite model for positive data clustering. The main motivation is the fact that this kind of data is naturally encountered in several applications from different domains and disciplines. We develop an efficient and principled fully Bayesian approach for the learning of the proposed infinite model. The results of applying the proposed model to the challenging problem of text categorization are encouraging. The proposed approach has a considerable variety of potential applications such as images categorization and gene expression data analysis. While surely powerful in mixture model's learning, MCMC techniques are unfortunately very time consuming. A possible solution, that we are working on actively, is the development of a variational approach to tackle the learning problem. Another potential future work could be the introduction of a feature selection component like the one proposed in [6] to improve further the modeling performance.

Acknowledgment. The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. McLachlan, G.J., Peel, D.: Finite Mixture Models. Wiley, New York (2000)
2. Bouguila, N., Ziou, D.: MML-Based Approach for Finite Dirichlet Mixture Estimation and Selection. In: Perner, P., Imiya, A. (eds.) MLDM 2005. LNCS (LNAI), vol. 3587, pp. 42–51. Springer, Heidelberg (2005)
3. Bouguila, N., Ziou, D.: On Fitting Finite Dirichlet Mixture Using ECM and MML. In: Singh, S., Singh, M., Apte, C., Perner, P. (eds.) ICAPR 2005. LNCS, vol. 3686, pp. 172–182. Springer, Heidelberg (2005)
4. Rasmussen, C.E.: The Infinite Gaussian Mixture Model. In: Advances in Neural Information Processing Systems (NIPS), pp. 554–560 (2000)
5. Biernacki, C., Celleux, G., Govaert, G., Langrognet, F.: Model-Based Cluster and Discriminant Analysis with the MIXMOD Software. Computational Statistics and Data Analysis 51, 587–600 (2006)
6. Boutemedjet, S., Ziou, D., Bouguila, N.: Unsupervised Feature Selection for Accurate Recommendation of High-Dimensional Image Data. In: Advances in Neural Information Processing Systems (NIPS), pp. 177–184 (2007)

7. Bdiri, T., Bouguila, N.: Learning Inverted Dirichlet Mixtures for Positive Data Clustering. In: Kuznetsov, S.O., Ślęzak, D., Hepting, D.H., Mirkin, B.G. (eds.) RSFDGrC 2011. LNCS, vol. 6743, pp. 265–272. Springer, Heidelberg (2011)
8. Antoniak, C.E.: Mixtures of Dirichlet Processes With Applications to Bayesian Nonparametric Problems. The Annals of Statistics 2(6), 1152–1174 (1974)
9. Dunson, D.B., Pillai, N., Park, J.-H.: Bayesian Density Regression. Journal of the Royal Statistical Society (B) 69(2), 163–183 (2007)
10. Duan, J.A., Guindani, M., Gelfand, A.E.: Generalized Spatial Dirichlet Process Models. Biometrika 94(4), 809–825 (2007)
11. Bouguila, N., Ziou, D.: A Dirichlet Process Mixture of Dirichlet Distributions for Classification and Prediction. In: Proc. of the IEEE Workshop on Machine Learning for Signal Processing (MLSP), pp. 297–302 (2008)
12. Bouguila, N., Ziou, D.: A Nonparametric Bayesian Learning Model: Application to Text and Image Categorization. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 463–474. Springer, Heidelberg (2009)
13. Rodriguez, A., Dunson, D.B., Gelfand, A.E.: Bayesian Nonparametric Functional Data Analysis Through Density Estimation. Biometrika 96(1), 149–162 (2009)
14. Bouguila, N., Ziou, D.: A Dirichlet Process Mixture of Generalized Dirichlet Distributions for Proportional Data Modeling. IEEE Transactions on Neural Networks 21(1), 107–122 (2010)
15. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6, 721–741 (1984)
16. Gelfand, A.E., Smith, A.F.M.: Sampling-Based Approaches to Calculating Marginal Densities. Journal of the American Statistical Association 85, 398–409 (1990)
17. Hastings, W.K.: Monte Carlo Sampling Methods using Markov Chains and their Applications. Biometrika 57, 97–109 (1970)
18. Tiao, G.G., Cuttman, I.: The Inverted Dirichlet Distribution with Applications. Journal of the American Statistical Association 60(311), 793–805 (1965)
19. Stoica, P., Selen, Y., Li, J.: Multi-Model Approach to Model Selection. Digital Signal Processing 14, 399–412 (2004)
20. Neal, R.M.: Markov Chain Sampling Methods for Dirichlet Process Mixture Models. Journal of Computational and Graphical Statistics 9, 249–265 (2000)
21. Gilks, W.R., Wild, P.: Algorithm AS 287: Adaptive Rejection Sampling from Log-Concave Density Functions. Applied Statistics 42(4), 701–709 (1993)
22. Joachims, T.: Estimating the Generalization Performance of an SVM Efficiently. In: Proc. of ICML, pp. 431–438 (2000)
23. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)

Multi-Label Weighted *k*-Nearest Neighbor Classifier with Adaptive Weight Estimation

Jianhua Xu

School of Computer Science and Technology,
Nanjing Normal University, Nanjing 210097, China
xujianhua@njnu.edu.cn

Abstract. Multi-label classification is a special learning task in which any instance is possibly associated with multiple classes simultaneously. How to design and implement efficient and effective multi-label algorithms is a challenging issue. The *k*-nearest neighbor (*k*NN) method and its weighted form (W*k*NN) are simple but effective for binary and multi-class classification. In this paper, we construct a weighted *k*NN version for multi-label classification (MLC-W*k*NN) according to Bayesian theorem. Through approximating a query instance by the linear weighted sum of *k*-nearest neighbors in terms of least squared error, the weights are determined adaptively by solving a quadratic programming with a unit simplex constraint. Specially, our MLC-W*k*NN is still a model-free and instance-based learning technique and only involves a tunable parameter *k*. Experimental study on two benchmark data sets (Image and Yeast) illustrates that our MLC-W*k*NN outperforms seven existing high-performed multi-label algorithms.

Keywords: Multi-label classification, *k*-nearest neighbor rule, Bayesian theorem, Weight estimation, Quadratic programming.

1 Introduction

Multi-label classification is an extension of traditional multi-class one, in which any instance could belong to several classes simultaneously and thus the classes are not mutually exclusive [1]. Nowadays, the existing discriminative multi-label approaches can be mainly categorized into two groups: algorithm extension and data decomposition. The former considers all instances and all classes at once, resulting into some complicated optimization problems, e.g., large-scale quadratic programming in multi-label support vector machine (Rank-SVM) [2] and unconstrained optimization problem in multi-label neural networks (BP-MLL) [3]. The latter implicitly or explicitly applies some data decomposition trick (e.g., one-versus-rest, one-versus-one or label powerset) to divide an entire multi-label data set into a series of binary or multi-class subsets which are easy and convenient to be solved by lots of existing single-label classifiers or their modified versions, e.g., multi-label naïve Bayesian classifier (ML-NB) [4] and radial basis function neural networks (ML-RBF) [5]. However, how to design and implement efficient and effective multi-label classifiers is still a challenging issue.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 79–88, 2011.
© Springer-Verlag Berlin Heidelberg 2011

The *k*-nearest neighbor (*k*NN) method [6-8] is one of the oldest and simplest algorithms for classical single-label (binary or multi-class) learning. Despite its simplicity, the asymptotic analysis shows that the error rate will not greater than twice the Bayesian error rate [6, 7]. In essence, such an original *k*NN method assumes that each neighbor has the same contribution to a query instance. The distance-weighted *k*NN method (W*k*NN) was proposed in [9], which weighs the instances nearby more heavily than those farther away. Some analytical weighting functions have been designed in [9-11], most of which monotonously decrease as the distance increases. The performance of W*k*NN has been verified by many experiments [10, 11].

Recently, some complicated methods were introduced to determine weights [12-14]. A query instance is approximated by a linear weighted sum of *k*-nearest neighbors through minimizing a regularized risk functional. The *k* weights are detected by linear ridge regression [12, 13], by quadratic programming [13], and by fast primal-dual log-barrier interior point method [14]. But, at least, an additional tunable parameter (i.e., regularization constant) has to be tuned in such three methods. Besides, it is possible that a few weights are negative in linear ridge regression for lack of non-negative constraints in [12, 13].

Three *k*NN-based multi-label classifiers have been proposed [1, 15, 16]. The simplest form (i.e., MLC-*k*NN in this paper) is a simple and direct extension of *k*NN, which counts the number of labels belonging to each class among *k*-nearest neighbors, and then detects relevant labels using the threshold *k*/2 for each class. Note that this method can be interpreted by one-versus-rest decomposition and binary *k*NN rule [1].

Two improved *k*NN-based multi-label methods (i.e., ML-*k*NN [15] and IBLR-ML [16]) cascade the *k*NN rule with a probabilistic model. The former uses discrete binary Bayesian formula, while the latter logistic regression. To estimate quantities in the second model, both of them need a training stage. Based on both leave-one-out (LOO) procedure and *k*NN rule, an original multi-label training set is converted into a new one, in which each instance is described by frequencies of labels occurring in *k*-nearest neighbors in ML-*k*NN and by weighted sum of positive and negative labels over *k*-nearest neighbors for each class in IBLR-ML. Then ML-*k*NN estimates class prior and conditional probabilities, while IBLR-ML calculates weights and bias in logistic functions, for each class independently. Therefore, strictly speaking, such two approaches are not model-free and instance-based yet.

In this paper, we still pursue model-free and instance-based characteristic of *k*NN rule and further do not introduce any new adjustable parameter. We extend the weighted *k*NN rule to construct its multi-label version (MLC-W*k*NN) according to Bayesian theorem. In terms of least squared error, a query instance is approximated by the linear weighted sum of *k*-nearest neighbors only, which results in a quadratic programming with one unit simplex constraint for detecting weights. Our MLC-W*k*NN only involves a tunable parameter *k*. Experimental results on two data sets (Image and Yeast) demonstrate that our MLC-W*k*NN outperforms seven existing multi-label algorithms.

This paper is organized as follows. In Section 2, we construct a novel MLC-W*k*NN method according to Bayesian theorem. Experimental results are reported in Section 3. Finally Section 4 concludes this paper.

2 A Novel Multi-Label Weighted *k*-Nearest Neighbor Algorithm

In this section, we first review traditional multi-class weighted *k*-nearest neighbor (*k*NN) method, and then build a novel multi-label weighted *k*NN algorithm as per Bayesian theorem and introduce an adaptive weight estimation method.

Let a *q*-class multi-label training data set of size *l* drawn identically and independently from an unknown probability distribution be,

$$\{(\boldsymbol{x}_1, \boldsymbol{y}_1), ..., (\boldsymbol{x}_i, \boldsymbol{y}_i), ..., (\boldsymbol{x}_l, \boldsymbol{y}_l)\}, \tag{1}$$

where $\boldsymbol{x}_i \in R^d$ denotes a *d*-dimensional instance vector, and $\boldsymbol{y}_i = [y_{i1}, ..., y_{ij}, ..., y_{iq}]^T$ is its *q*-dimensional binary label vector, in which $y_{ij} = +1$ if $\boldsymbol{x}_i$ belongs to the *j*th class (i.e., relevant label), otherwise $y_{ij} = -1$ (i.e., irrelevant label). For multi-label classification, some label vectors involve several positive label components. When any binary label vectors consist of one positive label component only, multi-label classification degenerates into traditional multi-class one.

2.1 Traditional Multi-class Weighted *k*NN Method

Given a query instance $\boldsymbol{x}$, we finds out the first *k*-nearest neighbors with some distance metric from the training set (1), i.e.,

$$\{(\overline{\boldsymbol{x}}_1, \overline{\boldsymbol{y}}_1), ..., (\overline{\boldsymbol{x}}_i, \overline{\boldsymbol{y}}_i), ..., (\overline{\boldsymbol{x}}_k, \overline{\boldsymbol{y}}_k)\}. \tag{2}$$

Further the *k* distances between $\boldsymbol{x}$ and $\overline{\boldsymbol{x}}_i (i = 1, ..., k)$ in increasing order are denoted by $d_1, ..., d_k$. For multi-class classification, the *k*NN rule counts the number of instances belonging to each class from (2) as its decision functions, i.e.,

$$f_j(\boldsymbol{x}) = |\{i \mid \overline{y}_{ij} = +1, i = 1, ..., k\}|, \; j = 1, 2, ..., q, \tag{3}$$

where $\overline{y}_{ij} = \pm 1$ is the *j*th binary label of the *i*th nearest neighbor. When each nearest neighbor $\overline{\boldsymbol{x}}_i$ is assigned to a non-negative weight $w_i \geq 0$, the W*k*NN rule calculates the sum of weights of instances belonging to each class as its decision functions

$$f_j(\boldsymbol{x}) = \sum_{i=1}^{k} \left(w_i \mid \overline{y}_{ij} = +1 \right). \tag{4}$$

Regardless of *k*NN and W*k*NN, the majority voting strategy is used to assign $\boldsymbol{x}$ to one of *q* classes only, i.e.,

$$y_j = \begin{cases} +1, if \; f_j(\boldsymbol{x}) \geq f_{j'}(\boldsymbol{x}), j' = 1, ..., q, j' \neq j, \\ -1, \qquad\qquad otherwise, \end{cases} \tag{5}$$

where $\boldsymbol{y} = [y_1, ..., y_j, ...y_q]^T$ denotes the binary label vector of $\boldsymbol{x}$, whose one component is +1 here.

2.2 Multi-Label Weighted *k*NN Method Based on Bayesian Theorem

For multi-label classification, the above formulae (3)-(5) have to be modified properly since an instance possibly has several positive labels at the same time. Now we consider the positive and negative labels of each class among *k*-nearest neighbors (2) for $\boldsymbol{x}$. For the *j*th class, the numbers of positive and negative labels are,

$$n_j^+ = \sum_{i=1}^{k} (\overline{y}_{ij} \mid \overline{y}_{ij} = +1), n_j^- = \sum_{i=1}^{k} (-\overline{y}_{ij} \mid \overline{y}_{ij} = -1), \tag{6}$$

and correspondingly two prior probabilities that $\boldsymbol{x}$ belongs to positive and negative labels respectively become

$$P_j(+1) = \frac{n_j^+}{k}, P_j(-1) = \frac{n_j^-}{k}. \tag{7}$$

For the simplest multi-label version of *k*NN, i.e., MLC-*k*NN, the *q* decision functions can be defined as follows,

$$f_j(\boldsymbol{x}) = \tfrac{1}{k}\sum_{i=1}^{k} \overline{y}_{ij} = \tfrac{1}{k}\sum_{i=1}^{k} (\overline{y}_{ij} \mid y_{ij} = +1) - \tfrac{1}{k}\sum_{i=1}^{k} (-\overline{y}_{ij} \mid y_{ij} = -1) = P_j(+1) - P_j(-1). \tag{8}$$

This expression represents the difference of prior probabilities between positive and negative labels for each class. In [13], it is shown that the normalized weights can be expressed as an adaptive kernel for density estimation. Thus, for our multi-label form of W*k*NN, i.e., MLC-W*k*NN, we define two conditional probabilistic densities that $\boldsymbol{x}$ belongs to positive and negative labels respectively for each class as follows,

$$p_j(\boldsymbol{x} \mid +1) = \sum_{i=1}^{k} (w_{ij} \mid \overline{y}_{ij} = +1), p_j(\boldsymbol{x} \mid -1) = \sum_{i=1}^{k} (w_{ij} \mid \overline{y}_{ij} = -1). \tag{9}$$

According to Bayesian theorem, the *q* decision functions are constructed as,

$$f_j(\boldsymbol{x}) = \frac{P_j(+1)p_j(\boldsymbol{x} \mid +1) - P_j(-1)p_j(\boldsymbol{x} \mid -1)}{P_j(+1)p_j(\boldsymbol{x} \mid +1) + P_j(-1)p_j(\boldsymbol{x} \mid -1)}, j = 1, ..., q. \tag{10}$$

This formula represents the difference of posterior probabilities between positive and negative labels for each class. Now, for multi-label classification, the predicted binary vector $\boldsymbol{y} = [y_1, ..., y_j, ..., y_q]^T$ of $\boldsymbol{x}$ is determined by the following rule for (8) and (10),

$$y_j = \begin{cases} +1, if\ f_j(\boldsymbol{x}) \geq 0 \\ -1,\ otherwise \end{cases} \tag{11}$$

From (8) and (10), it is observed that, among *k*-nearest neighbors, for each class, MLC-*k*NN only utilizes the prior information of positive and negative labels, while our novel MLC-W*k*NN the prior and conditional probabilistic information simultaneously.

2.3 Adaptive Weight Estimation Method

Some simple analytic weighting functions have been introduced in [9-11], most of which assign larger weights to instances nearby than to those farther away. In many real-world applications, some instances in a training set are often very similar, which possibly provides highly-correlated information and thus is not helpful for improving the performance of *k*NN-based classifiers. In [13], a query instance $\boldsymbol{x}$ is approximated by a linear weighted sum of *k*-nearest neighbors via minimizing a regularized risk functional, and then the weights vector is determined by solving a quadratic programming. In this paper, to avoid any additional adjustable parameter, we adopt a simplified approximation in terms of least squared error, i.e.,

$$\min\left(\boldsymbol{x}-\sum_{i=1}^{k} w_i\bar{\boldsymbol{x}}_i)\right)^T\left(\boldsymbol{x}-\sum_{i=1}^{k} w_i\bar{\boldsymbol{x}}_i\right),\ \text{s.t.}\sum_{i=1}^{k} w_i=1, w_i\geq 0, \tag{12}$$

which can further be converted into a standard quadratic programming with a unit simplex constraint,

$$\min \tfrac{1}{2}\boldsymbol{w}^T(\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}})\boldsymbol{w}-(\bar{\boldsymbol{X}}^T\boldsymbol{x})^T\boldsymbol{w},\ \text{s.t.}\boldsymbol{1}^T\boldsymbol{w}=1, \boldsymbol{w}\geq\boldsymbol{0}, \tag{13}$$

with $\bar{\boldsymbol{X}}=[\boldsymbol{x}_1,...,\boldsymbol{x}_k]$, $\boldsymbol{w}=[w_1,...,w_k]^T$, $\boldsymbol{1}=[1,...,1]^T$ and $\boldsymbol{0}=[0,...,0]^T$. In the above formulae (13), the linear term $\bar{\boldsymbol{X}}^T\boldsymbol{x}$ characterizes the similarities between the query instance $\boldsymbol{x}$ and any neighbor, and due to the negative sign a large weight would be allocated to the most similar neighbor. The quadratic term $\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}$ describes the similarities between any two neighbors, and thus two small weights would be assigned to two most similar neighbors and two large weights to two most dissimilar neighbors. In this paper, we solve (13) to detect weights adaptively for any query instance in our MLC-W*k*NN.

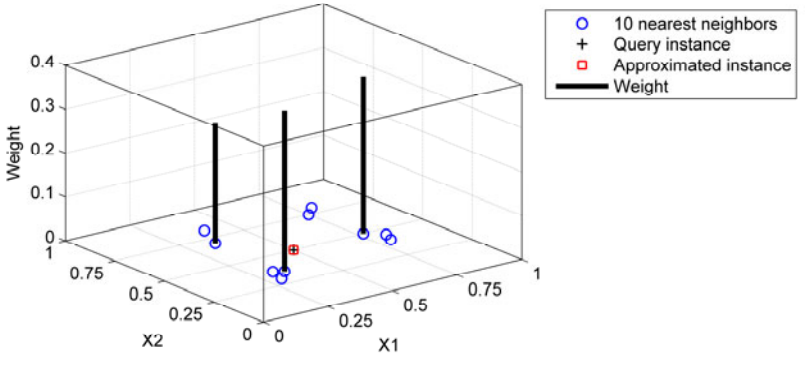


Fig. 1. A simple 2-dimensional example for adaptive weight detection

In Fig.1, a simple 2-dimensional example is illustrated, where 10 neighbors are clustered into four groups and the symbol cross denotes our query instance. Three large weights are assigned to three distinct instances by solving (13), in which one

instance is the most similar to the query instance, and two instances are the most dissimilar each other. It is also observed that the query instance is exactly approximated, as shown by the symbol square in Fig. 1.

3 Experiments

In this section, we report and analyze some experimental results with our MLC-W*k*NN and seven existing multi-label methods including three *k*NN-based approaches (MLC-*k*NN, ML-*k*NN [15], and IBLR-ML [16]) coded in C/C++ by us, and the other four techniques (Rank-SVM [2], BP-MLL [3], ML-NB [4] and ML-RBF [5]) whose Matlab software is from [17], on two famous data sets: Image and Yeast.

3.1 Five Evaluation Measures and Two Data Sets

It is more complicated to evaluate a multi-label classifier than a single one. In this paper, we utilize the same five popular and indicative measures as those in [3-5, 15, 16], as listed in Table 1. Here, the up arrow means that the higher some measure is, the better some method performs, and just the reverse for the down arrow. These five measures are calculated for a single test instance and then are averaged over all test instances. Among five measures, only the Hamming loss depends on the predicted label set of a test instance, while the other four are associated with the decision function values. On their detailed definitions, please refer to [3-5, 15, 16].

Table 1. Five measures used in our experiment

Measure	Abbreviation	Interval	Good performance
Coverage	Cov	$[0, q]$	↓
One error	1-Err	[0, 1]	↓
Average precision	A-Prec	[0, 1]	↑
Ranking loss	R-Loss	[0, 1]	↓
Hamming loss	H-Loss	[0, 1]	↓

In this paper, we collect two widely used benchmark data sets: Image and Yeast from [17, 18]. Table 2 provides their useful statistics, such as, the number of training instances, test instances, features and classes (or labels), and the average labels.

Table 2. Statistics for two data sets used in this paper

Data set	Training instances	Test instances	Features	Classes	Average Labels
Image	1200	800	294	5	1.24
Yeast	1500	917	103	14	4.24

The Image data set is a semantic scene classification task [15, 17], which consists of 2000 pictures. Each picture is first converted from RGB space into CIE LUV space, and divided into 49 blocks using 7x7 grids. Then the mean and variance of each block are calculated. In this case, 49x3x2=294 features are extracted. All pictures are identified as five conceptual classes: desert, mountain, sea, sunset and

tree. This set is partitioned into training and test subsets according to 60% versus 40% by us. The Yeast data set focuses on a gene function classification problem of yeast based on 2417 genes [2, 18]. Each gene is described using micro-array expression data and phylogenetic profile, from which 103 features are extracted. This set involves 14 gene functional classes and 4.24 of average labels, and has been divided into training and test subsets in [18].

3.2 Tuning *k* Value for Four *k*NN-Based Classifiers

For four *k*NN-based classifiers, there is only a tunable parameter *k* after the Euclidean distance has been used. We tune *k* value from 1 to 49 by steps of 2 using three-fold cross validation on the training sets of Image and Yeast. In order to relieve the possible effect of random seeds, three repeats are realized. Under this setting, we use the average values over three repeats as our parameter tuning indexes.

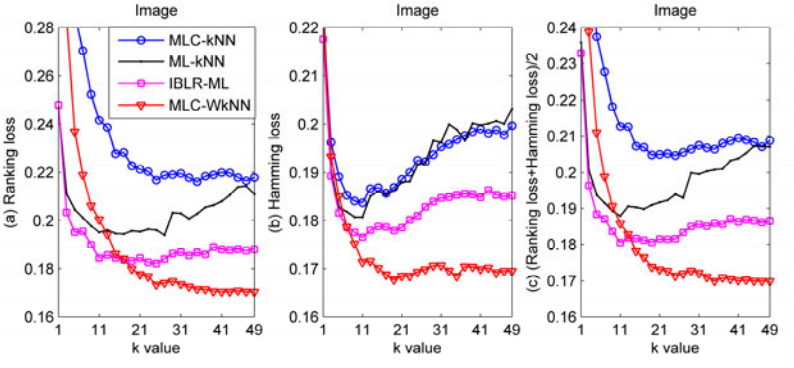


Fig. 2. The tuning procedure of *k* value for Image

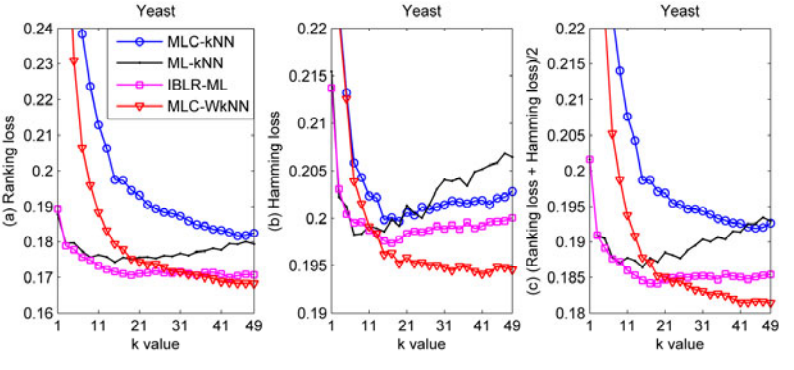


Fig. 3. The tuning procedure of *k* value for Yeast

To begin with, the (average) ranking and Hamming losses are calculated as a function of *k* value for Image and Yeast in Fig. 2 and 3 (a) and (b). It is observed that the ranking loss curves behave relatively smoothly, while the Hamming loss ones

irregularly. Obviously, the optimal k values from the minimum ranking loss are not consistent with those from the minimum Hamming loss. In order to consider the trade-off between such two indexes, we define their average value as our final criterion function to choose the optimal k value for all four kNN-based methods,

$$\text{Criterion function} = (\text{Ranking loss} + \text{Hamming loss})/2, \tag{14}$$

as shown in Fig. 2 and 3 (c) for Image and Yeast. It is found out that as the k value increases, (a) the curves of MLC-WkNN for two data sets and MLC-kNN for Yeast decrease almost monotonously first and then hold almost unchanged; (b) the curves of ML-kNN, IBLR-ML for two data sets and MLC-kNN for Image have a valley value respectively; (c) after the k value is greater than some constant (15 for Image and 25 for Yeast), our MLC-WkNN is consistently superior to the other three methods. Finally, the optimal k values and their minimal criterion values are detected and listed in Table 3 for such four methods on two data sets. It is observed that our MLC-WkNN and MLC-kNN need more neighbors to achieve an optimal criterion value than ML-kNN and IBLR-ML, and further our MLC-WkNN performs the best among four kNN-based methods, as shown in bold font in Table 3.

Table 3. The optimal k values and their criterion values for four methods

Data set	MLC-kNN	ML-kNN	IBLR-ML	MLC-WkNN
Image	25(0.2046)	11(0.1898)	11(0.1805)	35(**0.1700**)
Yeast	45(0.1919)	15(0.1864)	17(0.1842)	41(**0.1815**)

This tuning experiment not only searches for the optimal k value for four kNN-based methods, but also provides a primary comparison and evaluation between our MLC-WkNN and three existing kNN-based classifiers.

3.3 Comparison Study on Two Test Data Sets

According to the optimal k values in Table 3 for four kNN-based methods and the recommended parameter settings for the other four approaches in [2-5], we retrain these eight classifiers using two entire training sets and then verify their performance using two corresponding test sets listed in Table 2. The experimental results are reported in Tables 4 and 5, where the best measure value among eight methods is highlighted in bold font. In order to obtain a comprehensive comparison, we order these methods on each measure first and then calculate the average rank (i.e., A-Rank) of each method over five measures, as shown in the last columns of Tables 4 and 5. This comparison method of different classifiers was recommended in [19].

In Tables 4 and 5, our MLC-WkNN works the best on all measures but one error for Image, and all five ones for Yeast. Interestingly, for Image, ML-RBF performs the best on one error measure and as well as our MLC-WkNN on Hamming loss measure in Table 4. In Table 5, our MLC-WkNN, BP-MLL and IBLR-ML achieve the same smallest one error values for Yeast. In terms of average rank in the last columns in Table 4 and 5, our MLC-WkNN obtains a top position among eight methods. It is also illustrated that ML-kNN and IBLR-ML improve the performance of original MLC-kNN through adding posterior probability estimation, while our ML-WkNN boosts the

performance of MLC-*k*NN via weighted trick and adaptive weight estimation more effectively. Among the other four approaches, ML-RBF obtains a relatively stable performance, while BP-MLL, ML-NB and Rank-SVM behave slightly badly.

According to the above experiments and discussions, it can be illustrated that our MLC-W*k*NN is a competitive method for multi-label classification, compared with seven existing methods.

Table 4. Experimental results for Image

Method	Cov ↓	1-Err ↓	A-Prec ↑	R-Loss ↓	H-Loss ↓	A-Rank ↓
Rank-SVM	1.2163(7)	0.4563(7)	0.7113(7)	0.2406(7)	0.2383(8)	7.2
BP-MLL	1.6138(8)	0.5450(8)	0.6348(8)	0.3395(8)	0.2340(7)	7.8
ML-NB	1.0987(6)	0.3700(6)	0.7598(6)	0.2039(6)	0.1995(6)	6.0
ML-RBF	0.9175(4)	**0.2913(1)**	0.8069(2)	0.1620(3)	**0.1643(1)**	2.2
MLC-*k*NN	1.0500(5)	0.3388(5)	0.7729(5)	0.1940(5)	0.1873(5)	5.0
ML-*k*NN	0.9388(3)	0.3238(4)	0.7949(4)	0.1695(4)	0.1790(4)	3.8
IBLR-ML	0.9088(2)	0.3075(3)	0.8050(3)	0.1619(2)	0.1718(3)	2.6
MLC-W*k*NN	**0.8800(1)**	0.3013(2)	**0.8094(1)**	**0.1527(1)**	**0.1643(1)**	**1.2**

Table 5. Experimetal results for Yeast

Measure	Cov ↓	1-Err ↓	A-Prec ↑	R-Loss ↓	H-Loss ↓	A-Rank ↓
Rank-SVM	6.5082(6)	0.2497(7)	0.7241(8)	0.1960(8)	0.2214(8)	7.4
BP-MLL	6.4144(3)	**0.2366(1)**	0.7506(5)	0.1749(4)	0.2094(7)	4.0
ML-NB	6.7012(8)	0.2399(5)	0.7459(7)	0.1813(7)	0.2081(6)	6.6
ML-RBF	6.5878(7)	0.2334(4)	0.7539(4)	0.1784(6)	0.2011(4)	5.0
MLC-*k*NN	6.4482(5)	0.2541(8)	0.7496(6)	0.1780(5)	0.2027(5)	5.8
ML-*k*NN	6.4318(4)	0.2410(6)	0.7567(3)	0.1733(3)	0.1987(3)	3.8
IBLR-ML	6.3784(2)	**0.2366(1)**	0.7610(2)	0.1693(2)	0.1962(2)	1.8
MLC-W*k*NN	**6.2824(1)**	**0.2366(1)**	**0.7643(1)**	**0.1668(1)**	**0.1932(1)**	**1.0**

4 Conclusions

In this paper, we construct a multi-label weighted *k*-nearest neighbor algorithm according to Bayesian theorem. Given a query instance, the weights are detected by solving a quadratic programming problem with a unit simplex constraint. The experimental study on two data sets in terms of five evaluation measures shows that our MLC-W*k*NN outperforms seven existing multi-label methods, which are based on *k*-nearest neighbor rule, neural networks, naïve Bayes and support vector machine respectively. From our study in this paper, it can be concluded that the weighting trick with adaptive weight estimation is an effective tool to improve the performance of *k*NN-based multi-label methods.

In our future work, we will conduct a model selection work to search optimal *k* value more efficiently rather than *k*-fold cross validation and parameter scan, and test more large-scale data sets.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under grant no. 60875001. We also thank Prof. Zhou and Zhang, whose Matlab software is used in our study.

References

1. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. International Journal of Data Warehousing and Mining 3(3), 1–13 (2007)
2. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) The 14th Conference on Neural Information Processing Systems (NIPS 2001), pp. 681–687 (2001)
3. Zhang, M.L., Zhou, Z.H.: Multi-label neural networks with application to function genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering 18(10), 1338–1351 (2006)
4. Zhang, M.L., Pena, J.M., Robles, V.: Feature selection for multi-label naïve Bayes classification. Information Science 179(19), 3218–3229 (2009)
5. Zhang, M.L.: ML-RBF: RBF neural networks for multi-label learning. Neural Processing Letters 29(2), 61–74 (2009)
6. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)
7. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. Wiley, New York (1973)
8. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.H., Steinbach, M., Hand, D.J., Steinberg, D.: Top 10 algorithms in data mining. Knowledge and Information Systems 14(1), 1–37 (2008)
9. Duduni, S.A.: The distance-weighted k-nearest neighbor rule. IEEE Transactions on Systems, Man, and Cybernetics 6(2), 325–327 (1976)
10. Macleod, J.E.S., Luk, A., Titterington, D.M.: A re-examination of the distance-weighted *k*-nearest neighbor classification rule. IEEE Transactions on Systems, Man, and Cybernetics 17(4), 689–696 (1987)
11. Zavrel, J.: An empirical re-examination of weighted voting for k-NN. In: Daelemans, W., Flach, P., van den Bosch, A. (eds.) The 7th Belgian-Dutch Conference on Machine Learning, pp. 139–148 (1997)
12. Zuo, W., Zhang, D., Wang, K.: On kernel difference-weighted k-nearest neighbor classification. Patten Analysis and Applications 11, 247–257 (2008)
13. Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity-based classification: concepts and algorithms. Journal of Machine Learning Research 10, 747–776 (2009)
14. Gupta, M.R., Gray, R.M., Olshen, R.A.: Nonparametric supervised learning by linear interpolation with maximum entropy. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(5), 766–781 (2006)
15. Zhang, M.L., Zhou, Z.H.: ML-*k*NN: a lazy learning approach to multilabel learning. Pattern Recognition 40(7), 2038–2048 (2007)
16. Cheng, W.W., Hullermeier, E.: Combining instance-based learning and logistic regression for multi-label classification. Machine Learning 76(2/3), 211–225 (2009)
17. Zhang, M.L.: Image data set and multi-label learning software (BP-MLL, ML-NB, ML-RBF and Rank-SVM), `http://cse.seu.edu.cn/people/zhangml`
18. Tsoumakas, G.: Multi-label datasets, `http://mulan.sourceforge.net/datasets.html`
19. Brazdil, P.B., Soares, C.: A Comparison of Ranking Methods for Classification Algorithm Selection. In: Lopez de Mantaras, R., Plaza, E. (eds.) ECML 2000. LNCS (LNAI), vol. 1810, pp. 63–74. Springer, Heidelberg (2000)

Emotiono: An Ontology with Rule-Based Reasoning for Emotion Recognition

Xiaowei Zhang[1], Bin Hu[1,2,*], Philip Moore[2], Jing Chen[1], and Lin Zhou[1]

[1] The School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China
{zhangxw,bh,jchen10,zhoul_07}@lzu.edu.cn
[2] The School of Computing, Telecommunications and Networks, Birmingham City University, Birmingham B42 2SU, UK
{bin.hu,philip.moore}@bcu.ac.uk

Abstract. Recently, the field of automatic recognition of users' affective states has gained a great deal of attention. Automatic, implicit recognition of affective states has many applications, ranging from personalized content recommendation to automatic tutoring systems. In this work, we propose an ontology called 'Emotiono' for the robust recognition of emotions through Electroencephalogram (EEG). In 'Emotiono', we define entities such as users' emotions, EEG features and their relationships. With inference rules obtained by Decision Tree algorithm, users' current emotional state can be reasoned based on their EEG data. We implement 'Emotiono' in Protégé 4.1 and evaluate its performance with EEG data gathered from the eNTERFACE06_EMOBRAIN Database. Using a 9-fold cross validation method for training and testing, 'Emotiono' reaches an average classification rate of 97.80% for recognizing 5 subjects' emotional states.

Keywords: Ontology, EEG, Emotion, Decision Tree.

1 Introduction

Recently, concomitant with advances in hardware technology and signal processing techniques, diverse issues related to the content-based emotion retrieval, classification and recommendation have received significant traction. Intelligent context-aware systems adapt their behavior according to context [1]; a context may include: location, time, activity, physiological signals, a person's individual characteristics, and a user's affective state(s). The Semantic Web with ontology-based context-modeling using inference and reasoning has been shown to provide an effective basis for expressing the relationships between contextual elements and to infer new data [2]. However, the existing ontological context modeling frameworks are not capable of solving specific problems related to ontological affective modeling and little attention has been paid to physiological signals related to emotion recognition.

In our research, we are exploring the features of bio-signals by mapping them to emotion recognition parameters. Work conducted in previous psycho-physiology

* Corresponding author.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 89–98, 2011.
© Springer-Verlag Berlin Heidelberg 2011

research provides evidence of a strong relationship between physiological reactions and emotional/affective states of humans [3]. However, these features demonstrate serious limitations in the retrieval of semantic information. The goal of our research is to build an ontology-based context model for emotion recognition and enable reasoning in pervasive computing environments. In this paper we discuss the structure of our ontology with the concepts involved in emotion recognition. Specific reference is made to Electroencephalography (EEG) and users' affective states, such as one semantic relationship among a person, a subset of his/her EEG features and the related affective state. A person has an affective state when the EEG feature values are equal to specific numerical references. The conditions: e.g., a headache complicated by long exposure to a noisy environment and low light levels (in a room) may induce an affective state such as boredom.

This paper is organized as follows. The following section describes ontology-based context modeling (OBM) which expresses EEG features and emotion entities in a smart space environment with other relevant information. Section 3 briefly outlines the data used in the context-modeling and data processing. Section 4 sets out a context inference mechanism based on first-order logic. Section 5 presents case-based reasoning results using our 'Emotiono' ontology. The final section presents conclusions with potential topics for future work.

2 'Emotiono' Ontology Construction

The Semantic Web [4] is an extension of the current web in which information that can be processed by a machine coexists and complements human-readable information; the aim is to enable computers and people to work in cooperation in a more efficient way. Ontologies [5] figure prominently in the emerging Semantic Web as a way of representing the semantics of documents thus enabling these semantics to be used by web applications and intelligent software agents.

2.1 Affective Model Applied in the 'Emotiono' Ontology

According to a currently widely accepted view, a simple but yet effective theory employs a bidirectional emotional model defined by two basic parameters/variables: the affective degree of valence and arousal. The two-dimensional emotion plane can be divided into four quadrants (HVHA, HVLA, LVLA, and LVHA) as shown in Figure 1. Thus, each emotional state can be defined as some combination of the valence and arousal components [6] [7]. In this research, we use three different emotional classes: calm, exiting positive and exiting negative. By selecting several images from the IAPS (International Affective Picture System) according to the rules in equation (1), where the arousal and valence scales are from 1 to 9, we get three subsets of images that can be used for stimulation of three different emotional classes:

$$\begin{aligned} &calm: \ \overline{arousal} < 4; \ \ 4 < \overline{valence} < 6 \\ &positive\ \ exciting: \ \overline{valence} > 6.8; \ \ \overline{arousal} > 5; \ \ \ Var(valence) < 2 \\ &negative\ \ exciting: \ \overline{valence} < 3; \ \ \overline{arousal} > 5 \end{aligned} \tag{1}$$

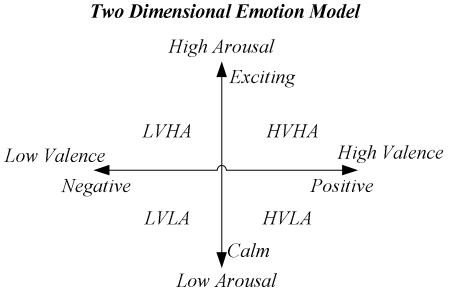


Fig. 1. Visualization of the emotion model. Pictures in the 2-D emotional space are defined by the valence and the arousal dimension.

2.2 The Structure of the 'Emotiono' Ontology

We have developed an emotional and related ontology in the Affective Computing domain. The W3C Web Ontology Language (OWL) [8] has been used to represent the ontology created using the Protégé 4.1 editor [9]. OWL enables domain representation by defining classes and related properties of those classes to describe individuals and their characteristics such as name, age, and other personal details that describe a person's 'context'. Describing individuals using their context provides a basis upon which it is possible to reason about these classes and individuals. Figure 2 shows a representation of some 'Emotiono' ontology classes with their definitions and properties. The 'Emotiono' ontology defines the terms used to describe and represent emotional knowledge including basic concepts (different affective states, participants' EEG features and sampling rate) and the relationships that exist among them. Some concepts and some hierarchy of concepts in the 'Emotiono' ontology with their subclasses include: Spatial_Parameter, Channel_Type and Electrode.

Each class has attributes that define the region the class occupies. The class (or classes) that the property applies to is termed (in OWL terms) the property's domain, while the predicate and the related data types and their literal values the property can take is called the property's range. In our ontology, we deal with two properties [10]:

(1) Object properties (objectProperty): 'hasEmotion', 'onCondition'; these properties carry information relating to the type of the 'Context'. The Context can be a Subject, an EEG_Recording_Parameter, or a Biosignal. These properties are created and defined using OWL object properties. This approach enables one resource (OWL Class) to have a defined relationship (with constraints where applicable) with another resource (OWL Class).

(2) Datatype properties (datatypeProperty): 'nameValue'; such properties define the subject <S> (the Class), the predicate <P>, and the object <O> (a literal value). The literal value represents the data value being processed. The literal value may indicate for example the name of the user being studied.

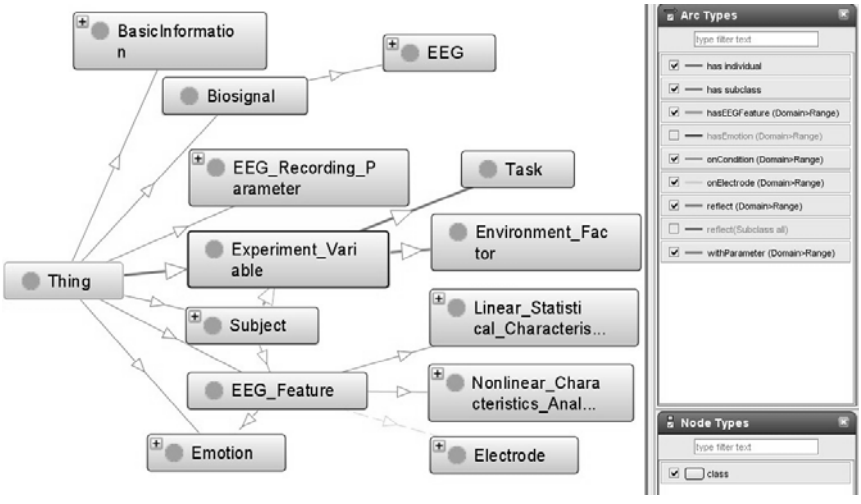


Fig. 2. Partial view of the 'Emotiono' ontology

3 Data Processing Method

It is well known that the variation of the surface potential distribution on the scalp reflects functional and physiological activities emerging from the underlying brain [11]. We get an individual's emotional information by analyzing his EEG features derived from the raw EEG data.

3.1 Data Collection

In this study, data collected from the sixth eNTERFACE workshop [12] are used. The EEG data were collected from five subjects. The subjects carried out three different mental tasks, calm, exciting positive and exciting negative, while watching images from the IAPS that corresponded to the three emotional classes. After each stimulus, there was a black screen for 10 seconds, and the participant was asked to give a self-assessment of his emotional state.

3.2 Data Preprocessing and EEG Features

Initially, the raw EEG signals are prepared for use in a preprocessing stage. From a simple comparison between the self assessments and the expected values from the IAPS database, we found that some stimuli did not evoke the expected emotions. For some of the stimuli, the participants noted that they felt really different. Apparently these stimuli are not clear enough to raise certain emotions on the participants. For that reason we do not use stimuli for which

$$\in \text{valence} = |\overline{\text{selfassessmentvalence}} - E(\text{valence})| > 1 \quad \text{or}$$

$$\in \text{arousal} = |\overline{\text{selfassessmentarousal}} - E(\text{arousal})| > 1 \ .$$

This resulted in the removal of samples, then 40 trials from the first and second subjects and 63 trials from the rest subjects are used in our research.

A Bandpass filter is used to smooth the signals and eliminate EEG signal drifting and EMG disturbances; a Wavelet Algorithm eliminates EOG disturbances. The raw signals (obtained from five subjects) are trimmed to a fixed time length of 12 seconds. Features are extracted by sliding 4s windows with a 2s overlap between consecutive computations. Typical statistical values such as the mean value and standard deviation, linear and nonlinear measures are computed on 54 channels. Overall 1300 EEG features are extracted from all electrodes.

4 Rule-Based Reasoning

In 'Emotiono' ontology, a user's desired emotional state is deduced from the ontology based on his situation (prevailing state), personal information, and his EEG features. In order to get the main relations between EEG features of a certain person and his affective states Generic rule reasoner, a Jena Reasoner engine is used; the reasoner consists of the reasoning engine and context-based engine. The context-based engine extracts the contexts of interrelation with input data for emotion recognition. Therefore, the 'Emotiono' ontology relies on well-defined context definitions to arrive at the correct emotional state. When the reasoner receives the EEG signal data or user request, a context-based reasoning engine generates the query as rules to generate the correct results.

4.1 The Reason of Generating Rules by C4.5

Inference rules are based on a number of EEG features. For EEG feature extraction, researchers have investigated many methods including: frequency domain analysis, the combination of different features extracted from the frequency domain and the cross-correlation between electrodes. Most of these features (like frequency domain, time domain and statistical analysis) are computed in our research. A large number of EEG features, served as knowledge related to affective states, are built into the ontology and are expressed as concepts or individuals. A decision list received from a decision tree is a set of "IF-THEN" statements. In our research, the subject's EEG features are routed down the decision tree according to the values of the attributes in successive nodes. When a leaf is reached a rule is generated according to the specific emotion assigned to that leaf.

The C4.5 algorithm [14] (one type of Decision Tree) is used in our research to generate rules. The motivation for this selection includes:

(1)The C4.5 algorithm merely selects features which are most relevant to differentiate each affective state;

(2)The C4.5 algorithm is a rule-based reasoning method and is searched sequentially for an appropriate if-then statement to be used as a rule.

The Reasoner can deduce the emotional state using a correspondence of a small number of EEG features/rules, thus enhancing inference speed. The C4.5 algorithm has been used effectively in a number of documented research projects to achieve

accurate emotion classification [15]. Based on the results reported in the literature we have also applied the C4.5 algorithm classification technique to the 'Emotiono' ontology. The output takes the form of a tree and classification rule which is a basic knowledge representation style that many machine learning methods use [16].

C4.5 is used as a predictor by 9-fold cross-validation on the data sets. Following complete creation of the tree it should be pruned. This process is designed to reduce classification errors caused by specialization in the training set, and update the data set by removing features which are less important.

4.2 Emotion Recognition Rules

We have identified the most significant EEG features and reasoning rules using the C4.5 algorithm so that the avoidance of redundant rules has been achieved. The EEG features for the five subjects are used for generating rules. The result is achieved using the J48 classifier (a Java implementation of C4.5 Classifier) in the Waikato Environment for Knowledge Analysis (WEKA). The confidence factor used for pruning is set at [C = 0.25], whereas the minimum number of instances per leaf is set at [M = 2]. The accuracy of the decision tree is measured by means of a 9-fold cross validation.

Variables in reasoning rules represent the resources (subjects, situations, EEG features) which are found using SPARQL [17] queries run on the 'Emotiono' ontology. The RDF model descriptions and rules in the demonstration are serialized in XML/RDF (as defined in the 'Emotiono' OWL file produced by the Protégé 4.1 editor). Identification of the emotional state becomes the static pattern involving the dynamic combination of the EEG features and selection of necessary information from the current situation. A rule, with its "IF-THEN" structure defines a basic fact about user's current emotional state. An example of the rules is depicted as follow:

```
String rules = "[Rule1: (?subject rdf:type base:Subject)
(?EEG_feature1 rdf:type ? Beta/ Theta)
(?EEG_feature1 base:hasValue ?value1) lessThan(?value1, 2.3)
(?EEG_feature1 base:onElectrode ?electrode1)
(?electrode1 rdfs:label "CP4")
(?EEG_feature2 rdf:type ? Beta/Theta)
(?EEG_feature2 base:hasValue ?value2) lessThan(?value2, 1.7)
(?EEG_feature2 base:onElectrode ?electrode2)
(?electrode2 rdfs:label "FT8") (?EEG_feature3 rdf:type ? Ppmean)
(?EEG_feature3 base:hasValue ?value3) lessThan(?value3, 2.5)
(?EEG_feature3 base:onElectrode ?electrode3)
(?electrode3 rdfs:label "TP8") (?emotion rdf:type base:Emotion)
(?emotion base:hasSymbol "1") ->
(?subject base:hasEmotion ?emotion)] ".
```

The corresponding tree is depicted in Figure 3. Shown is the routing down the tree according to the arrow, and when the leaf is reached one rule is generated according to the calm assigned to the leaf.

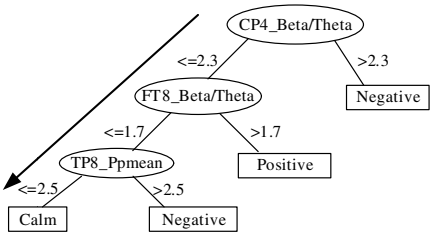


Fig. 3. A simple Rule-based Decision Tree giving a reasoning on emotions

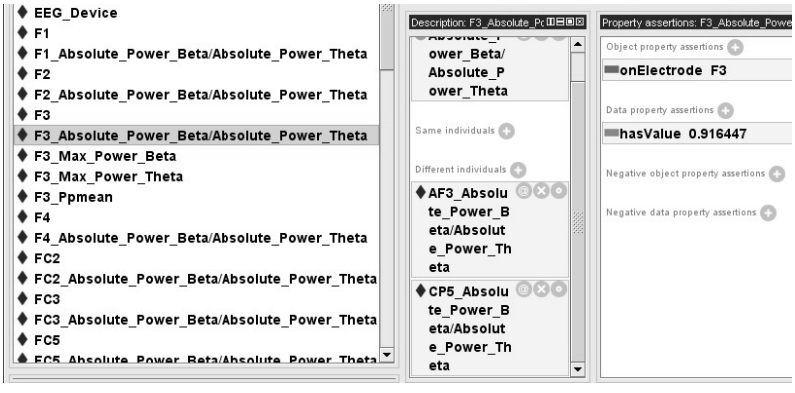


Fig. 4. Part of subject1's information

5 Reasoning Results

We have taken the information for the first subject (marked as subject1) as the test data to be used in the 'Emotiono' ontology. The user's basic information (Age, Gender) and 1300 EEG features are written into Emotiono ontology. Examples of the data used in the ontology are shown in Figure 4.

The data is then inputted into inference engine and the user's affective state (Positive) is deduced. This process is graphically modeled in Figure 5.

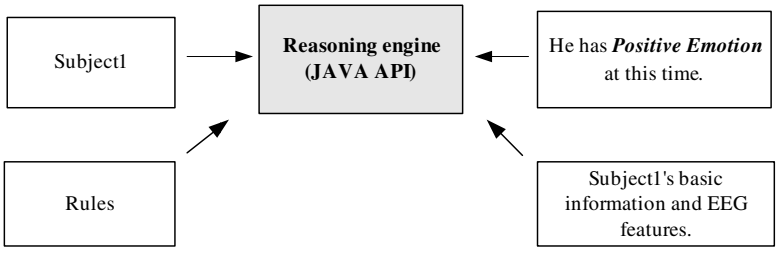


Fig. 5. Reasoning on subject1's affective state

In the example, we have an EEG feature dataset for subject1 which 'Emotiono' annotates with the following values: (1) Asymmetry_Alpha_F4/F3 = 1.78, (2) O2_Skewness = 2.36, and (3) P3_Ppmean = 4.67, etc. This point is classified by means of the ontology under the positive emotional concept.

The EEG features for five subjects (whose raw EEG data come from the sixth eNTERFACE workshop) were inputted into BP neural network and 'Emotiono' ontology. Although both of these approaches can recognize and classify the affective emotional states, the accuracy of classification is quite different as can be seen from Table1.

Table 1. The accuracy of BP neural network and 'Emotiono' ontology

	Sample size	The accuracy of classification using the ontology	The accuracy of classification using BP neural network
subject1	200	100%	70.50%
subject2	200	98.50%	76.50%
subject3	315	96.51%	70.76%
subject4	315	100%	66.33%
subject5	315	93.97%	75.54%
average		97.80%	71.93%

We find that emotions of subject1 and subject4 are classified correctly by means of 'Emotiono' Ontology. Other data also resulted in an improved level of emotional recognition using the ontology approach as compared with the results obtained using the classifier BP neural network classifier.

6 Conclusions and Future Work

The principal contribution of our approach is the ability to define emotion information, the subject's EEG data related to emotions, and situations at the level of concepts as they apply to the OWL class(s). Not only do we specify the uncertainty of concept's value (property's value) but also specify uncertain relationships between concepts by inference. Since ontologies mainly deal with concepts within a specific domain, our context model can easily extend the current ontology-based modeling approach. Based on our research into human emotions and physiological signals, we have defined a human emotion-oriented context ontology which captures both logical and relational knowledge. Given the context ontology, we can potentially combine the 'Emotiono' ontology with other knowledge bases which address similar applications. For example, we can use it in a health care domain for treatment on mental and emotional disorders. Additionally, we can add information inferred by EEG features into an existing ontology by adding relations, relation chains and restrictions without constructing a new ontology. Thus, our work into context modeling supports scalability and knowledge reusability. Since properties or restrictions of classes in 'Emotiono' are implicitly defined in the ontology and reasoning rules are derived

from the mapping-relations between nodes in C4.5, the mapping process can be programmed to run automatically. This feature provides a basis for the reduction of the burden on knowledge experts and developers when compared to previously documented research [18] [19]. Since rules between EEG features and different affective states are formed, we can easily extend from reasoning to learning about uncertain context, which is simply mapping about the rules and nodes of C4.5.

This paper describes our approach of representing and reasoning about uncertainty and context. Our study presented in this paper shows that the proposed context model is feasible and necessary for supporting context modeling and reasoning in pervasive computing. Our work is part of an ongoing research into ubiquitous Affective Computing for pervasive Systems. However, dealing with a great mass of EEG data, reasoner takes a long running time, so we should shorten it and supply quicker data process speed in future work. In addition, we are also planning to update the dataset with increased number of subjects and scheduling to test different methodologies with increased number of data sets to get the most efficient one. Accordingly, we are exploring methods of integrating multiple reasoning methods from the AI field with their supporting representation mechanism(s) into the context reasoning.

Acknowledgement. This work was supported by the National Basic Research Program of China (973 Program) (grant No. 2011CB711001), National Natural Science Foundation of China (grant No. 60973138, 61003240), the EU's Seventh Framework Programme OPTIMI (grant No. 248544), and the Fundamental Research Funds for the Central Universities (grant No. lzujbky-2011-k02, lzujbky-2011-129).

References

1. Baldauf, M., Dustdar, S., Rosenberg, F.: A Survey on Context-Aware systems. International Journal of Ad Hoc and Ubiquitous Computing 2(4), 263–277 (2007)
2. Deborah, L.M., Frank, V.H.: OWL Web Ontology Language Overview W3C Recommendation (2004), http://www.w3.org/TR/owl-features
3. Ratner, C.: A Cultural-Physiological Analysis of Emotions. Culture and psychology 6, 5–39 (2000)
4. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (May 2001), issue
5. Chandrasekaran, B., Josephson, J.R., Benjamins, R.: What Are Ontologies and Why Do We Need Them. IEEE Intelligent Systems 14, 20–26 (1999)
6. Russell, J.A.: A circumplex model of affect. J. Pers. Soc. Psychol. 39(6), 1161–1178 (1980)
7. Watson, D., Tellegen, A.: Toward a consensual structure of mood. Psychol. Bull. 98(2), 219–235 (1985)
8. W3C, Web Ontology Language (OWL), http://www.w3.org/2004/OWL/
9. Protégé (ed.), http://protege.stanford.edu/
10. Antoniou, G., van Harmelen, F.: Web Ontology Language: OWL. In: Handbook on Ontologies in Information Systems, pp. 67–92 (2003)
11. Khalili, Z., Moradi, M.H.: Emotion Recognition System Using Brain and Peripheral Signals: Using Correlation Dimension to Improve the Results of EEG. In: International Joint Conference on (IJCNN 2009), pp.1571–1576 (2009)

12. The eNTERFACE06_EMOBRAIN Database, http://enterface.tel.fer.hr/docs/database_files/eNTERFACE06_EMOBRAIN.html
13. Frantzidis, C.A., Bratsas, C., Klados, M.A., Konstantinidis, E., Lithari, C.D., Vivas, A.B., Papadelis, C.L., Kaldoudi, E., Pappas, C., Bamidis, P.D.: On the Classification of Emotional Biosignals Evoked While Viewing Affective Pictures: An Integrated Data-Mining-Based Approach for Healthcare Applications. IEEE Transactions on Information Technology in Biomedicine, 309–314 (2010)
14. Quilan, R.J.: C4.5: Programs for Machine Learning. Morgan Kauffman, San Mateo (1993)
15. Jena Semantic Web Toolkit: http://www.hpl.hp.com/semweb/jena2.htm
16. Gu, T., Pung, H.K., Zhang, D.Q.: A Bayesian approach for dealing with uncertain contexts. Hot Spot Paper, Second International Conference on Pervasive Computing (Pervasive 2004), Vienna, Austria (2004)
17. SPARQL tutorial, http://www.w3.org/TR/rdf-sparql-query/
18. Ranganathan, A., Al-Muhtadi, J., Campbell, R.H.: Reasoning about Uncertain Contexts in Pervasive Computing Environments. IEEE Pervasive Computing 3(2), 62–70 (2004)
19. Wu, J.L., Chang, P.C., Chang, S.L., Yu, L.C., Yeh, J.F., Yang, C.S.: Emotion Classification by Incremental Association Language Features. Proceedings of World Academy of Science, Engineering and Technology 65, 487–491 (2010)

Parallel Rough Set: Dimensionality Reduction and Feature Discovery of Multi-dimensional Data in Visualization

Tze-Haw Huang[1], Mao Lin Huang[1], and Jesse S. Jin[2]

[1] School of Software, University of Technology Sydney,
Sydney 2007, Australia
tze-haw.huang@student.uts.edu.au, maolin@it.uts.edu.au
[2] School of Design, Communication and Information Technology,
University of Newcastle, Newcastle 2308, Australia
jesse.jin@newcastle.edu.au

Abstract. Attempt to visualize high dimensional datasets typically encounter over plotting and decline in visual comprehension that makes the knowledge discovery and feature subset analysis difficult. Hence, reshaping the datasets using dimensionality reduction technique is paramount by removing the superfluous attributes to improve visual analytics. In this work, we applied rough set theory as dimensionality reduction and feature selection methods on visualization to facilitate knowledge discovery of multi-dimensional datasets. We provided the case study using real datasets and comparison against other methods to demonstrate the effectiveness of our approach.

Keywords: Dimensionality Reduction, Rough Set Theory, Feature Selection, Knowledge Discovery, Parallel Coordinate, Visual Analytics.

1 Introduction

The effectiveness of visualization used to support knowledge discovery typically decline by a large number of dimensions. Dimensionality reduction is commonly used to address such problem that widely applied in mining the datasets to facilitate feature selection and pattern recognition. *Principal Component Analysis* (PCA) [1], *Multi-Dimensional Scaling* (MDS) [2] and *Self-Organizing Map* (SOM) [3] are the well-known unsupervised dimensionality reduction methods. They are efficient in projecting the dataset into low dimension space. However, the use of unsupervised methods on correlated dataset might produce unintuitive result due to minimal user influence to the algorithms. On the other hand, the supervised methods [4] usually require the user to define a set of weights known as threshold so the selection criteria would prefer dimensions for those weights above pre-defined threshold. For example, outliers are conceptually easy by finding a variance beyond the threshold but the quantization of outliers and its thresholds are difficult [5]. Although, it provides more intuitive and correlated result via user guidance but its efficiency greatly depends on the quantization of the weight of variables that is typically not a trivial task.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 99–108, 2011.
© Springer-Verlag Berlin Heidelberg 2011

The motivation of this work is to address the issues of 1) visual efficiency of Parallel Coordinate [6] by dimensionality reduction to enhance knowledge discovery 2) possibly non-intuitive result produced by unsupervised method that often being criticized as information loss 3) non trivial task of quantization in supervised method and 4) lack of support of feature discovery for multi-dimensional dataset in visualization.

In this paper, we proposed the Parallel Rough Set (PRS) visualization system that tightly integrated the Rough Set Theory (RST) with parallel coordinate visualization to facilitate knowledge discovery. The most distinct advantage of applying RST as a supervised dimensionality reduction is the concept of *condition* and *decision*. User simply specifies a dimension as decision and rest become conditions so the dimensions are reduced in such as way that fully respects to user specified decision.

2 Rough Set Theory Background

2.1 Classic Rough Set

RST was first introduced by Pawlak [7] in the field of approximation to classify objects in a set and in general it is applicable to any problems that require classification tasks. Given a dataset, let U be the finite set of objects called universe and A be the superset of all attributes $A = \{a_1, a_2, a_3, \ldots\ldots a_N\}, \exists! \, a_i \in A$ such that $a: U \to V_a, \ \forall a \in A$ where V_a is called the domain of a. A is further classified into two disjoint attribute subsets D called the decision attribute and rest the condition attributes C such that $A = (C, A \cup \{D\}), C \cap D = \emptyset$. For any objects $X \in U$ with non-empty subset $P \subseteq C \cup D$ are said to be discernible with respect to P if and only if the following equivalence relation is true:

$$f_P(X_i, X_j) = \begin{cases} 1, if \ V_a(X_i) = V_a(X_j), \forall V_a \in P, for \ i \neq j \\ 0, otherwise \end{cases} \quad (1)$$

Clearly, given the equivalence relation defined in (1) we can construct the equivalence classes denoted as $U/IND(P) = \{E_1, E_2, \ldots, E_n\}$ by partitioning U into disjoint subsets with the following indiscernibility relation:

$$IND(P) = \{(X_i, X_j) \in U : f_P(X_i, X_j) = 1\} \quad (2)$$

RST further defines three regions of approximation called lower approximation $\underline{P}X$, upper approximation $\overline{P}X$ and boundary region to approximate subsets $X \subseteq U$. The lower approximation and upper approximation also called positive and negative region respectively. The lower approximation contains objects that are surely in X and the upper approximation consists of objects that cannot be classified to X whereas the boundary region contains objects that possibly belong to X.

2.2 Variable Precision Rough Set

RST was initially designed to deal with consistent dataset by its strict definition of approximation regions. It assumes the underlying dataset is consistent with complete certainty of classifying objects into correct approximation regions. For example, if

$ab \rightarrow D$ then $cd \rightarrow D$ is considered as conflicting. This assumption of error-free classification of consistent dataset is unrealistic in most real world datasets. Although, a dataset can be partitioned into consistent and inconsistent data space and operates RST on the consistent one but we considered this is meaningless and unpractical use case. To deal with inconsistent dataset, Ziarko [8] argued that partially incorrect classification should be taken into account and hence proposed the Variable Precision Rough Set (VPRS) model as an inconsistent dataset extension to RST. VPRS model allows the probability classification by introducing a precision value β to relax the strict classification in original RST. It introduces the concept of *major inclusion* to tolerate the inconsistent dataset and the definition of *majority* implies no more than 50% of classification error so the admissible range of β is $(0.5, 1.0]$. The β positive in VPRS model are defined as:

$$POS_P^{\beta}(D) = \cup_{\Pr(IND(D)^*|X_i)\geq\beta} \{X_i \in IND(P)\} \tag{3}$$

Where $IND(D)^*$ and X_i denotes a set of the equivalent classes for D and $P \subseteq C$ respectively. Clearly, a portion of objects with specified value β in the equivalence classes need to be classified into decision class for it to be included in the β positive region. Ziarko also formulated the definition for *quality of classification* that is used to extract the β reducts and we will explain the definition of *reduct* in next section:

$$\gamma^{\beta}(P,D) = \frac{\left|\cup_{\Pr(IND(D)^*|X_i)\geq\beta} \{X_i \in IND(P)\}\right|}{|U|} where\ \Pr(P|D) = \frac{|D\cap P|}{|P|} \tag{4}$$

Where $|*|$ denotes the cardinality for the union of all the equivalence classes in the β positive region where classification is possible at specified value β with respect to relation $IND(D)^*$ and $|U|$ denotes the cardinality of the universe. Obviously, the quality of classification provides the measure for the degree of attribute dependency in such a way that if $\gamma^{\beta}(P,D) = 1$ means D fully depends on P at specified β value.

3 Parallel Rough Set System

PRS system consists of data model based on RST and visualization model based on parallel coordinate. In this section the incorporation of RST to achieve the dimensionality reduction and feature selection in dataset will be explained. We also used the classification method to reorder the dimensions to improve the visual structure of parallel coordinate.

3.1 Dimensionality Reduction via VPRS

The objective of dimensionality reduction in PRS is to employ VPRS to eliminate the superfluous dimensions by finding the optimal subset that is minimal yet sufficient to support the data exploratory analysis. There are certain advantages of using RST over other methods such as PCA, 1) it minimizes the impact of information loss by removing the irrelevant or dispensable dimensions and 2) the resultant subset of attributes is more intuitive by preserving the quality of classification. Typically we may find several subsets of attributes that satisfy the criteria called *reduct sets* denoted as

$R = \{P: P \subseteq C\}$. The minimal cardinality in the reduct sets called the minimal reduct denotes as R_{min} where $R_{min} \in C$ is the minimum subset of the condition attributes that cannot be reduced anymore while preserving the quality of classification with respect to decision attribute.

In VPRS model, the reduct is called β-reduct denoted as $RED^{\beta}(C, D)$ and according to Ziarko that a subset $P \subseteq C$ is a reduct of C with respet to D if and only if the following two criteria are satisfied:

1. $\gamma^{\beta}(C, D) = \gamma^{\beta}\left(RED^{\beta}(C, D), D\right)$ and,
2. No attributes can be eliminated from $RED^{\beta}(C, D)$ without affecting the requirement (1).

The requirement (2) can also be mathematically expressed as $POS^{\beta}_{P-\{a\}}(D) \neq POS^{\beta}_{P}(D), a \in P$. Obviously, Ziarko has defined the strict satisfaction of β reduct in requirement (1) that some attributes could only be removed if and only if its qualification of classification γ^{β} for subset $P \subseteq C$ must be the same against γ^{β} for whole set of original attributes C.

3.3 Feature Discovery via Rule Induction

Rule induction is also an important concept of RST and PRS took the advantage of it to support feature discovery on reduct. Typically, a rule is expressed as $X \rightarrow D_i$ in RST that learned from approximating a set of equivalent classes with respect to decision attribute using (3). In fact, the approximation regions used to determine β-reduct essentially act as rule templates where the equivalent classes classified into positive region will be the certain rules whereas the equivalent classes classified into boundary or negative region would be uncertain or negative rule respectively. We are interesting in the certain rules and need to highlight the importance of studying the rules because they enable the feature discovery of the dataset. For example, given a rule $X = \{wight_{high}, accel._{low}\} \rightarrow 80\%\ cylinder_{high}$ means we are eighty percents confident that the cars with higher weight and lower acceleration usually have more cylinders from the given dataset. Surely, such information is very useful for dataset exploratory analysis. There are two characteristics associates with a rule (1) accuracy and (2) coverage [9]. Given a rule its accuracy is defined as:

$$accuracy(X \rightarrow D_i) = \frac{|X \cap D_i|}{|X|} \tag{5}$$

Where X denotes the equivalent class of condition attributes. The accuracy measures the strength of a rule with respect to D_i if its accuracy is below than β called a weak rule that is not significant and too weak to be meaningful. Similarly, the coverage of a rule can be measured by:

$$coverage(X \rightarrow D_i) = \frac{|X \cap D_i|}{|D_i|} \tag{6}$$

The coverage measures the generality of a rule with respect to a certain class in D. In general, a rule with higher accuracy does not necessary imply a lower coverage rule [10] and vice versa.

3.4 Dimension Reorder to Enhance Visual Structure

The overall visual structure of the parallel coordinate is susceptible to the order of dimensions because inappropriate order creates visual clutter by non-uniform line crossing as a side effect. The existing technique developed to arrange the dimensions is based on similarity measurement [11]. Interestingly, if similarities of adjacent dimensions are maximized based on shortest distance i.e. Euclidean distance, then the sum of distance of hypotenuses would be minimized. Hence, the global visual structure of lines tends to be leveled. In generally, there is no widely acceptable method of dimension reordering in information visualization.

In this work, we used the cardinality based method to reorder the dimension with aim to maximize the uniform line crossing along with color brushing to reveal overall visual structure. The following describes the steps:

1. For each dimension computes the cardinality by applying equation (2) and inset them to a list in ascending order. In RST, this step is essentially computing the equivalent class of a dimension.
2. Create an empty list, insert an entry from sorted list for dimension with highest cardinality and immediately follow by inserting an entry with lowest cardinality.
3. Repeat step 2 until the sorted list is empty.

Figure 1 provides the comparison of visualization with and without dimension ordering. Clearly, dimension reordering reveals greater visual structure.

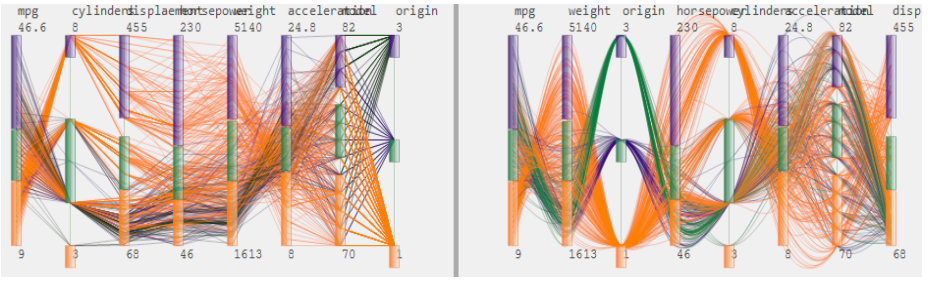


Fig. 1. (Left) Parallel coordinate using default dimension ordering (Right) Dimensions reordered using cardinality method which shows the better visual structure

4 Case Studies Using PRS

We would study the applications of PRS on two datasets obtained from StatLib, Carnegie Mellon University for dimensionality reduction and feature discovery. Both datasets were inconsistent.

The wage dataset consists of 11 attributes and 534 samples for the population survey in 1985. The attributes cover the sufficient information to describe the characteristics of a worker such as sex, wage, years of education, years of work experience, occupation, region of residence, race background, marital status and union membership. We first selected the experience as our decision target with β value sets to be 0.70 arbitrarily which simply instructs the system that our tolerance of classification

error with respects to experience is 70%. The system has reduced the dimensions from 11 to 6 and the result has shown in figure 2 where we could visually interpret that the people with more work experience tend to be older age, male and working in various sectors whereas people with less work experience has younger age and prefer to work in sectors other than construction and manufacturing.

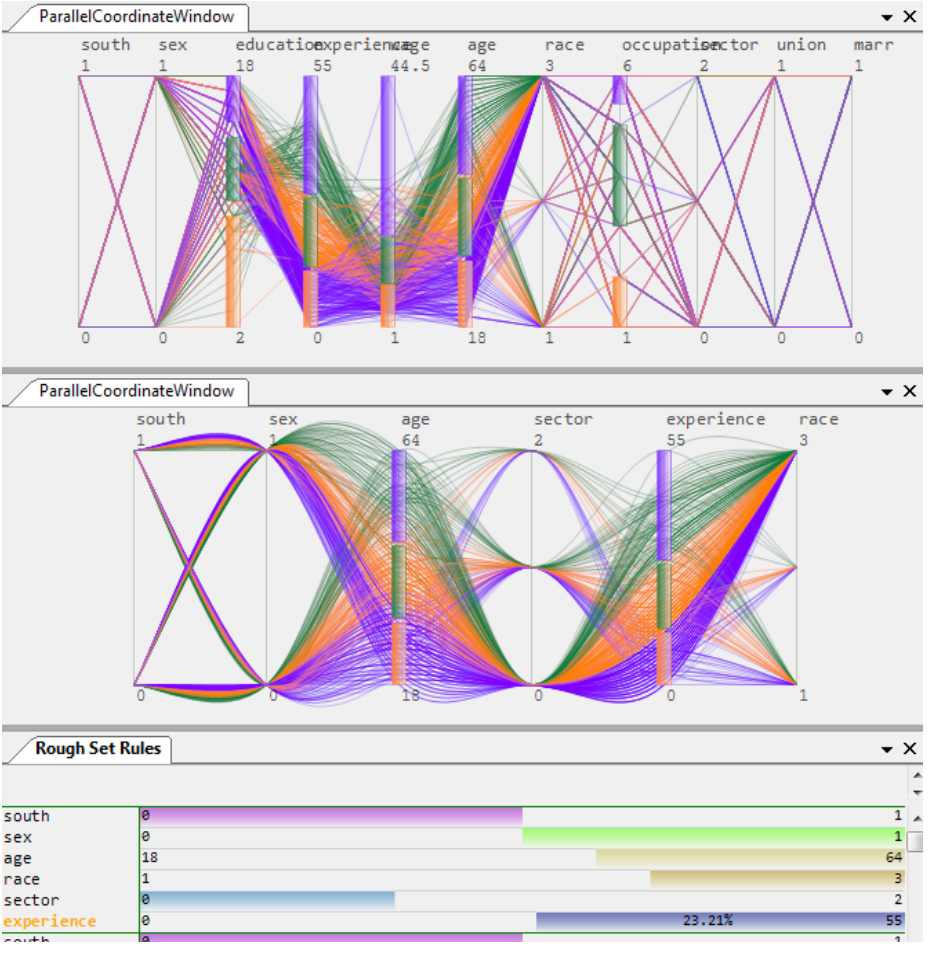


Fig. 2. (Top) Complete wage dataset visualization in parallel coordinate with dimensions reordered. (Middle) Dimensions reduced from 11 to 6 with 'experience' selected as decision. (Bottom) Feature discovery contains a set of rules derived. The bar indicates the value ranges and first rule has 23.21% coverage.

To further understand the interesting features from reduced dimensions we performed the feature discovery analysis that also illustrated in figure 2. The features were listed from most to least coverage of rules. It can be seen that first strongest rule has 23.21% of coverage about male lived not in south area with older age and work in non construction and manufacturing sectors typically has more work experience.

The second dataset used contains 8 attributes with 392 samples after removed the missing attribute objects. The dataset describes the car information about its origin, model, acceleration, weight, horsepower, cylinder, mileage per gallon (mpg) and displacement. We selected cylinders as decision attribute with β value set to 70% and the system has reduced the dimensions from 8 to 4. In figure 3 displayed the result of our operations.

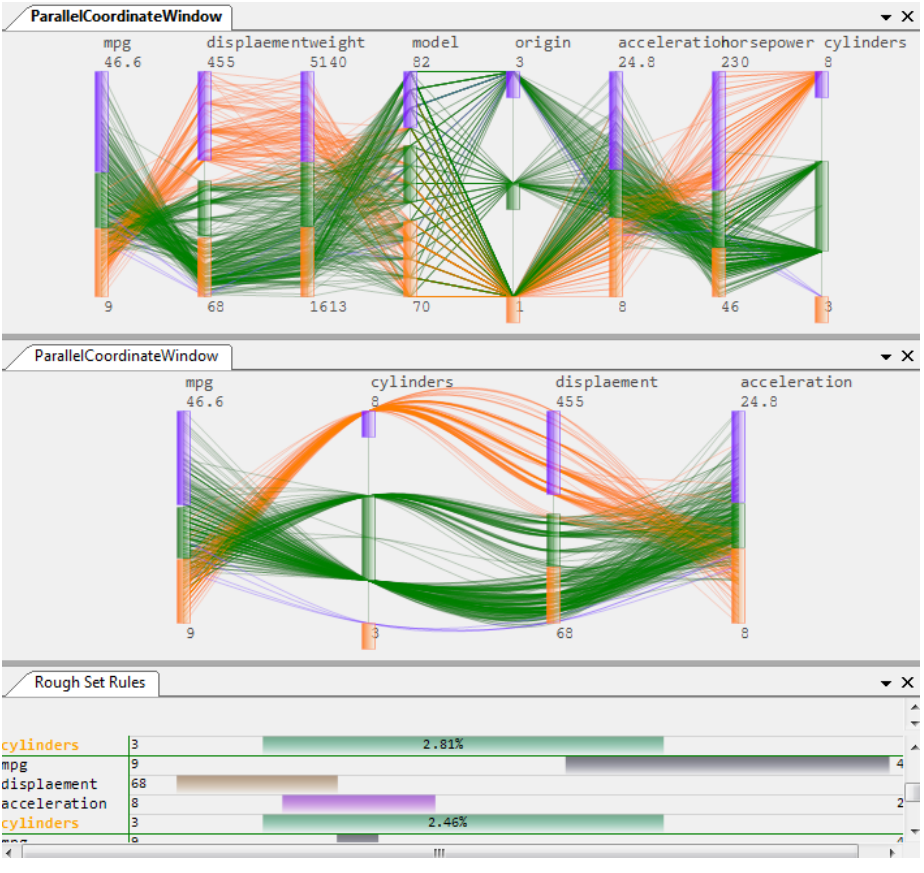


Fig. 3. (Top) Complete car dataset visualization in parallel coordinate with dimensions reordered. (Middle) Dimensions reduced from 8 to 4 with 'cylinders' selected as decision indicated. (Bottom) Feature discovery contains a set of rules derived from reduct.

Basically, the stronger the rule then the feature is usually more common sense. For example, the strongest rule derived indicates that 69.9% of cars with low mpg, high displacement and low acceleration typically equipped with more cylinders. Surely, it makes sense because cars with more cylinders consume more petrol and hence lower mileage per gallon. Therefore, we studied the weak rules in attempt to find interesting features. In figure 3 showed a weak rule that only has 2.46% coverage revealed cars

equipped 4~6 cylinders run higher mpg with relatively lower displacement and acceleration. Basically, these cars were poorly performed because cars with better mpg typically lighter and should possess higher acceleration.

Through the case studies, we demonstrated the powerful capabilities of PRS to support knowledge discovery. Traditionally, feature discovery requires experienced data analyst with domain knowledge in order to construct a complex SQL query. PRS as a visualization system is ease of use for user to focus on data subset via dimensionality reduction and to discover their features derived.

5 Comparison with Dimensionality Reduction Techniques

Comparison with PCA. Mathematically, PCA performs the orthogonal linear transformation that maps data to a low dimension space with non trivial computation of covariance matrix and eigenproblems. Since the value ranges for dimensions do not scale uniformly so we applied z-score standardization for each dimension on the car dataset. The z-score standardization is expressed as:

$$Z_i = \frac{x_i - \overline{x}}{\sigma} \; where \; \sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{x})^2}{(N-1)}} \tag{7}$$

Two most commonly used selection criterions in PCA were applied to select the principal components. The Kaiser criterion [12] is one of commonly acceptable criterion that simply ignores the components with eigenvalues less than one. Obviously, it is not applicable since the result is not intuitive for visualization with only one attribute qualified the criterion. Another popular criterion is Scree test proposed by Cattell [13] who suggested by plotting the eigenvalues on the graph to find the smooth decrease of eigenvalues then cut off the line and retains the components on the left side. Hence, with this guideline the selected attributes were origin and model by referring to figure 4. The disadvantage of using PCA in information visualization is the result might not be intuitive because the operations were carried out without considering any user inputs hence often being criticized as information loss.

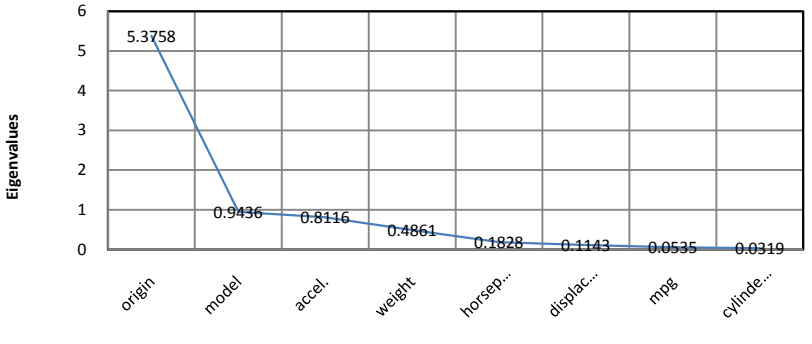


Fig. 4. Computed eigenvalues for each dimension on car dataset

Comparison with User-Defined Quality Metric (U-DQM). The similar supervised approach to allow user influence was introduced by Johansson et.al [4] where user-defined weighted combinations of quality metrics such as Pearson correlation, outlier and cluster detection are used to determine the dimensions to retain.

As a supervised dimensionality reduction, PRS made no assumption about the user knowledge which only requires the decision attribute as an user input and β value as tolerance for classification quality with respects to the decision attribute whereas in U-DQM the perquisite knowledge required to quantify the quality metric values might need greater user expertise. For example, the user needs to define the correlation, outlier and cluster value in such a way to avoid the insignificant correlations, outliers and clusters adding up to a sum that appears to be significant. Quantization is always difficult and not a trivial task where in U-DQM the recommendation values for correlation, outlier and cluster quality metrics are 0.05~0.5, 1 and 0.02 respectively in order to avoid large numbers of insignificant values appears to be significant. However, there is no clear benchmark of how these values were derived and probably in different datasets with different data types a value of 0.02 might not be appropriate. In terms of user input, we used percentage based for β whereas in U-DQM used absolute value with inconsistent scale for different quality metric that surely pose the challenges to the users. One of most important tasks of dimensionality reduction is the selection criteria for dimensions. The selection criterion of PRS is based on strict criteria defined by Ziarko where an attribute can be removed if and only if its removal does not affect the quality of classification against the whole set of attributes whereas in U-DQM it manually asks the user for the percentages of information loss that they are willing to sacrifice that obviously raises the challenge to the user again. In table 1 provided the use case summary between PRS and U-DQM.

Based on these empirical observations, classification based method employed by PRS provides more intuitive result than existing dimensionality reduction methods when deals with information correlated multi-dimensional dataset. This statement is based on fact that lacks of concept of decision attribute, surely, other algorithms could not guarantee about users concerned dimension in mind will be retained. Whereas, in PRS it guarantees that dimension will be retained known as decision and others will be removed if they are superfluous with respect to it. In addition, as a supervised method it does not expose excessive parameters to the user that typically requires quantization which is always difficult.

Table 1. Comparison summary between PRS and U-DQM

Comparisions	PRS	U-DQM
Information loss	Classification error	User sacrificed
User input	β, decision attribute	Quality metrics
Decision concept	Yes	No
Value input	%	Absolute value
Value scale	Uniform	Not uniform for each quality metric
Challenge	Define β for classification error	Quantifies values for various metrics

6 Conclusion

In this work, we contributed a novel PRS to facilitate knowledge discovery and data subset analysis for multi-dimensional dataset. The technique is based on the incorporation of RST and parallel coordinate. Surely, the concept of decision attribute provided is the most distinct feature than any existing methods in the field. Also, we were first to apply RST as dimensionality reduction and feature selection in visualization to the best of our knowledge. In the future work, we would like to further enhance PRS visual display such as dynamic decision tree, to support decision oriented knowledge discovery and such application is useful on medical related datasets.

References

1. Fodor, I.K.: A survey of dimension reduction techniques. Technical Report. UCRL-ID-148494. Lawrence Livermore National Lab., 1-18 (2002)
2. Kruskal, J.B., Wish, M.: Multidimensional scaling. Sage Publications, Beverly Hills (1977)
3. Kohonen, T.: The self-organizing map. Neurocomputing 21(1-3), 1–6 (1998)
4. Johansson, S., Johansson, J.: Interactive dimensionality reduction through user-defined combinations of quality metrics. IEEE Transaction on Visualization and Computer Graphics 15(6), 993–1000 (2009)
5. Choo, J., Bohn, S., Park, H.: Two-stage framework for visualization of clustered high dimensional data. In: Proc. of IEEE Symposium on VAST, pp. 67–74 (2009)
6. Inselberg, A.: The plane with parallel coordinates. The Visual Computer 1(2), 69–91 (1985)
7. Pawlak, Z.: Rough Set: Theoretical aspects of reasoning about data. Kluwer, Netherlands (1991)
8. Ziarko, W.: Variable precision rough set model. J. Comp. & Sys. Sci. 46(1), 39–59 (1993)
9. Tsumoto, S.: Accuracy and Coverage in Rough Set Rule Induction. In: Alpigini, J.J., Peters, J.F., Skowron, A., Zhong, N. (eds.) RSCTC 2002. LNCS (LNAI), vol. 2475, pp. 373–380. Springer, Heidelberg (2002)
10. Yao, Y., Zhao, Y.: Attribute reduction in decision-theoretic rough set models. Information Sciences 178(1), 3356–3373 (2008)
11. Ankerst, M., Berchtold, S., Keim, D.A.: Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In: Proc. of IEEE Symposium on Information Visualization, pp. 52–60 (1998)
12. Saporta, G.: Some simple rules for interpreting outputs of principal components and correspondence analysis. In: Proc. of ASMDA 1999. University of Lisbon (1999)
13. Cattell, R.B.: The scree test for the number of factors. Multivariate behavioral research 1(2), 245–276 (1966)

Feature Extraction via Balanced Average Neighborhood Margin Maximization

Xiaoming Chen[1,2], Wanquan Liu[2], Jianhuang Lai[1], and Ke Fan[2]

[1] School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, China
[2] Department of Computing, Curtin University Perth 6102, Australia

Abstract. Average Neighborhood Margin Maximization (ANMM) is an effective method for feature extraction, especially for addressing the Small Sample Size (SSS) problem. For each specific training sample, ANMM enlarges the margin between itself and its neighbors which are not in its class (heterogeneous neighbors), meanwhile keeps this training sample and its neighbors which belong to the same class (homogeneous neighbor) as close as possible. However, these two requirements are sometimes conflicting in practice. For the purpose of balancing these conflicting requirements and discovering the side information for both the homogeneous neighborhood and the heterogeneous neighborhood, we propose a new type of ANMM in this paper, called Balance ANMM (BANMM). The proposed algorithm not only can enhance the discriminative ability of ANMM, but also can preserve the local structure of training data. Experiments conducted on three well-known face databases i.e. Yale, YaleB and CMU PIE demonstrate the proposed algorithm outperforms ANMM in all three data sets.

Keywords: Feature Extraction, Balance ANMM, Face Recognition.

1 Introduction

Feature extraction is an attractive research topic in pattern recognition and computer vision. It aims to learn the optimal discriminant feature space to represent the original data. The feature space is usually a low-dimensional space in which the data's discriminant information is maintained and the redundant information is discarded. The processing of high-dimensional data generally needs unacceptable computational costs and this is known as the curse of high dimensionality. Moreover, the redundant information may cause classification deficiency. Therefore, feature extraction has become a significant preprocess step in many practical applications. In the past few decades, the methods for feature extraction such as Principle Component Analysis (PCA) [1] and Linear Disciminant Analysis (LDA) [2] have been widely applied in appearance-based face recognition and index-based document and text categorization, in which the data are usually represented by high-dimensional vectors.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 109–116, 2011.
© Springer-Verlag Berlin Heidelberg 2011

PCA is a popular unsupervised method. It performs feature extraction by seeking the directions in which the variances of the projected data in feature space are maximized. The low-dimensional space derived by PCA is efficient for representing the data, but it could not extract the discriminative information for classification since PCA does not consider any class labels of the data. LDA is a supervised method for learning a feature space to represent class separability. LDA enlarges the distances between the means of different classes meanwhile forces the data in the same class close to their mean. However, LDA generally suffers from three major drawbacks. Firstly, in the case of the Small Sample Size (SSS) problem [3][4], the within-class scatter matrix would be singular, so its inverse matrix does not exist. Secondly, LDA assumes the distribution of the data in each class is Gaussian distribution with a common variance matrix. Moreover, the class empirical mean is used as its expectation, and these assumptions may not be satisfied in practice. Thirdly, given a set sampled from c different classes, LDA can only extract c-1 dimensional feature at most, this may not produce the optimal solution.

To tackle these issues, various types of LDA are proposed [7][8][14][11] and recently the Average Neighborhood Margin Maximization (ANMM) is proposed in [5]. For a specific data sample, ANMM focuses on the difference between the average l_2 norm of this sample and its heterogeneous neighbors (the neighbors which have different class labels from this sample) and the average l_2 norm of this sample and its homogeneous neighbors (the neighbors which have the same class labels with this sample) in the feature space. Though as shown in [5], the performance of ANMM is better than some traditional methods, it still has three problems: firstly, ANMM only takes the information of class labels into account, but it does not preserve the intra-class or inter-class local structure in terms of the different "similarities" between the reference point and its neighbors. The issue of local structure preserving has been discussed in LPP [12], it is necessary to preserve the instinct local structure after projecting the data into a low dimensional subspace from the high dimensional data manifold, so that the discrimnant information can be remained [13]. Secondly, the l_2 norm between a specific sample and its heterogeneous neighbors is usually larger than the l_2 between it and its homogeneous neigbhors. Hence, the inter-class relationship is dominant in determining the projective map in ANMM. Thirdly, the small negative eigenvalues of $S-C$ imply that the heterogeneous neighbors are almost as close to the reference sample as the homogeneous neighbors. In other words, the margins for these two neighborhoods in this case are not differentiable. A good method of feature extraction needs to enlarge such ambigous margins but ANMM ignored them.

To overcome the drawbacks of ANMM, we propose a Balanced Average Neighborhood Margin Maximization (BANMM) in this paper. Three contributions are summarized as follows:

– we introduce the concept of *side information* and take it into ANMM so that different homogeneous neighbors or heterogeneous neighbors can be distinguished in terms of various similarities with the reference sample. The

relationship between different samples are redefined, which contains the local structure of the data set. Therefore, in the feature space, the locality can be preserved.
– A penalty parameter is adopted to maintain the discriminant information in the case that the margins of neighborhoods are ambiguous.

The rest of this paper is organized as follows: a brief review of ANMM is given in section 2. In section 3, the Balanced ANMM is introduced. The experimental results on face databases are shown in section 4. Section 5 is the conclusion.

2 Average Neighborhood Margin Maximization

ANMM aims to project the data into a feature space in which each data point can get close to its neighbors with the same class labels and separate from other points from different classes simultaneously. First, we present two key definitions in ANMM: *Homogeneous Neighborhood*: for a data point x_i, its ξ nearest homogeneous neighborhood $\mathcal{N}_i^o$ is the set of ξ most similar data which are in the same class with x_i; *Heterogeneous Neighborhood*: for a data point x_i, its ζ nearest Heterogeneous neighborhood $\mathcal{N}_i^e$ is the set of ζ most similar data which are not in the same class with x_i.

Based on these two definitions, the *average neighborhood margin* γ_i for each x_i is defined as

$$\gamma_i = \frac{1}{|\mathcal{N}_i^e|} \sum_{k:x_k \in \mathcal{N}_i^e} ||y_i - y_k||^2 - \frac{1}{|\mathcal{N}_i^o|} \sum_{j:x_j \in \mathcal{N}_i^o} ||y_i - y_j||^2 \tag{1}$$

where $y_i = W^T x_i$ is the image of x_i in the projected space and $|\cdot|$ is the cardinality of a set.

For each data point, formula (1) measures the difference two average l_2 norms in the feature space, the former one is the average l_2 norm of the image of x_i and the images of the data points in its heterogeneous neighborhood, the latter one is the average l_2 norm of the image of x_i and the images of the data points in its homogeneous neighborhood. By maximizing the total average neighborhood margin $\sum_i \gamma_i$, ANMM can push the data points which are not in the same class with x_i away and pull the data points which have the same class labels as x_i towards x_i. In this case, the ANMM criterion can be derived as follows:

$$\begin{aligned}\gamma = \sum_i \gamma_i = tr\{W^T [\sum_i \frac{1}{|\mathcal{N}_i^e|} \sum_{k:x_k \in \mathcal{N}_i^e} (x_i - x_k)(x_i - x_k)^T \\ - \sum_i \frac{1}{|\mathcal{N}_i^o|} \sum_{j:x_j \in \mathcal{N}_i^o} (x_i - x_j)(x_i - x_j)^T W]\} = tr[W^T (S - C) W]\end{aligned} \tag{2}$$

where $S = \sum_i \frac{1}{|\mathcal{N}_i^e|} \sum_{k:x_k \in \mathcal{N}_i^e} (x_i - x_k)(x_i - x_k)^T$
and $C = \sum_i \frac{1}{|\mathcal{N}_i^o|} \sum_{j:x_j \in \mathcal{N}_i^o} (x_i - x_j)(x_i - x_j)^T$. So with the constraint of $W^T W = I$, ANMM criterion becomes

$$\max_{W} tr\{W^T(S-C)W\} \quad s.t. W^T W = I \tag{3}$$

ANMM solves the optimization problem (3) by the Lagrangian method. The optimal projection matrix consists of the p eigenvectors corresponding to the largest p positive eigenvalues of S - C.

3 Balanced Average Neighborhood Margin Maximization

3.1 Side Information

In order to preserve the locality of original data and distinguish the different samples in homogeneous neighborhood and heterogeneous neighborhood, we first introduce the concept of "Side Information". The Side information represents the information that exists in the data set, which can be used to determine whether individual samples come from the same class or not, even the sample labels are not given. Side information has been discussed and applied in metric learning [17] [18]. Motivated by this concept, we define *similar neighborhood* (SN) and *dissimilar neighborhood* (DN) for an individual sample x_i in Balanced ANMM as follows:

$$SN_{x_i} = \{x_j | S_{ij} > \epsilon\} \cap \{x_j | x_j \in homogenerous\ neighborhood\ of\ x_i\} \tag{4}$$

$$DN_{x_i} = \{x_j | S_{ij} > \epsilon\} \cap \{x_j | x_j \in heterogenerous\ neighborhood\ of\ x_i\} \tag{5}$$

where S_{ij} represents the similarity of x_i and x_j, it can be Gaussian similarity or cosine similarity. ϵ is a threshold to control the similarity between x_i and its neighbors. Based on these two definitions, we adopt the similarity of a specific sample and its neighbors as a weight for calculating the relationship between them. The heavy weights are added on the neighbors of x_i which are closer to it than other neighbors in its similar neighborhood, so that BANMM can maintain them close to each other in the feature space, simultaneously the heavy weights are also given to the closer neighbors of x_i in its heterogenerous neighborhood in order to force their mapped points to seperate from the mapped point of x_i. Hence, the relationship between two individual samples in BANMM is defined as follow:

$$r(x_i, x_j) = ||x_i - x_j||^2 S_{ij} \tag{6}$$

where S_{ij} is the similarity between x_i and x_j. In this paper, the cosine similarity is adopted as $S_{ij} = \frac{|x_i^T x_j|}{||x_i|| ||x_j||}$.

3.2 BANMM

For a data set, it is obvious that the l_2 norms of a specific sample and its homogeneous neighbors are generally less than the ones of this sample and its heterogeneous neighbors. In this case, the latter should be more dominant in the

objective function of the optimization problem in ANMM. Hence, the Balanced Average Neighborhood Margin Maximization method adopts a positive balance parameter β to enhance the weight of intra-class relationship. The objective function of BANMM for a specific sample x_i is as follow:

$$J_i(W) = \beta \sum_{x_k \in DN_{x_i}} \frac{||W^T x_i - W^T x_k||^2 S_{ik}}{|DN_{x_i}|} - \sum_{x_j \in SN_{x_i}} \frac{||W^T x_i - W^T x_j||^2 S_{ij}}{|SN_{x_i}|} \quad (7)$$

where $|\bullet|$ is the cardinality of a set. Considering the total samples in the training data set, the objective can be defined as:

$$J(W) = \sum_i J_i(W) = tr\{W^T(\beta \sum_i \sum_{x_k \in DN_{x_i}} \frac{(x_i - x_k)(x_i - x_k)^T S_{ik}}{|DN_{x_i}|} - \sum_i \sum_{x_j \in SN_{x_i}} \frac{(x_i - x_j)(x_i - x_j)^T S_{ij}}{|SN_{x_i}|})W\} = tr[W^T(\beta \hat{S} - \hat{C})W] \quad (8)$$

where $\hat{S} = \sum_i \sum_{x_k \in SN_{x_i}} \frac{(x_i - x_k)(x_i - x_k)^T S_{ij}}{|DN_{x_i}|}$ and $\hat{C} = \sum_i \sum_{x_j \in SN_{x_i}} \frac{(x_i - x_j)(x_i - x_j)^T S_{ij}}{|SN_{x_i}|}$

For the purpose of increasing the weight of small eigenvalues of $\beta\hat{S} - \hat{C}$ in deriving the projective map, we import a penalty item into formula (8) in the final objective funtion of BANMM. In order to tune the parameter easily, we choose $1 - \beta$ as the coefficient of the penalty item, so the final optimization problem of BANMM is defined as:

$$\max_W tr\{W^T[\beta\hat{S} + (1-\beta)I - \hat{C}]W\} \quad s.t. W^T W = I \quad (9)$$

where I is the unit matrix. The optimal projective axes w_1,w_2,...,w_l can be selected as the eigenvecotrs corresponding to the l largest eigenvalues λ_1,λ_2,...,λ_l, i.e.,

$$[\beta\hat{S} + (1-\beta)I - \hat{C}]w_q = \lambda w_q, q = 1, 2, ..., l \quad (10)$$

where $\lambda_1 \geq \lambda_2, ..., \geq \lambda_l$. So far, we obtain the optimal projective matrix W of BANMM.

BANMM is an extension of ANMM in the follows:

1. For a specific sample x_i, the homogenerous neighborhood and the heterogenerous neighborhood are replaced by the similar neighborhood and disimilar neighborhood, since we consider to exploit the side information and preserve the local structure in the original data set.
2. The balance parameter β is introduced in the objective funtion of BANMM to balance the weights of inter-class relationship and intra-class relationship in learning the projective map.
3. The penalty term is used in the objective funtion of BANMM to enlarge the weight of small eigenvalues of $\beta\hat{S} + (1-\beta)I - \hat{C}$, so that the the ambigous margins cannot be ignored any more.

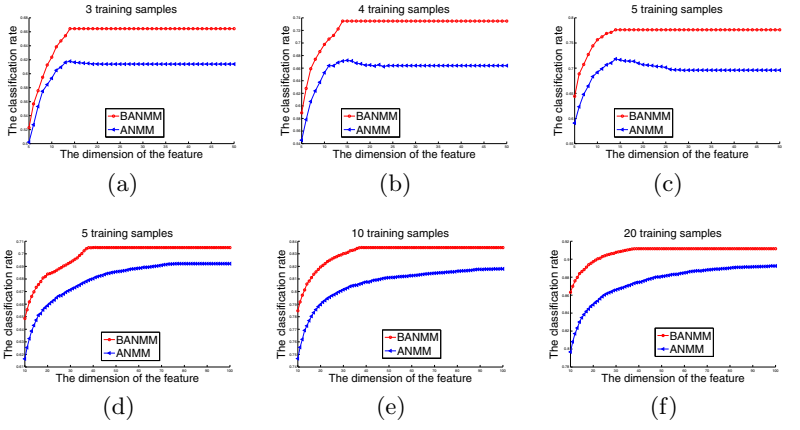


Fig. 1. (a)-(c) are the face recognition rates on YaleB database with 3, 4, 5 training samples for each person. (d)-(f) are the face recognition rates on YaleB database with 5, 10, 20 training samples for each person.

4 Experimental Results

In this section, we present the performance of the proposed BANMM method for the discriminant information extraction. As a new version of ANMM, BANMM is compared with ANMM as a method for feature extraction in face recognition. Three well-known face databases are chosen as benchmarks: Yale, YaleB and CMU PIE. The face databases are preprocessed to locate the face. Each image is normalized (in scale and orientation) and cropped into 32×32. The nearest neighbor is adopted as the classifier in all the experiments. Since in [5], it has shown that the performance of ANMM was better than some tranditional methods PCA [6], LDA(PCA + LDA) [3], MMC [10], SNMMC[15] and MFA [16]. Moreover, LPP [12] is a special case of MFA [16], so we only compare the proposed method with ANMM and choose PCA + LDA as the baseline in this paper. We randomly select i, ($i = 3, 4, 5$ for Yale database, $i = 5, 10, 20$ for YaleB database and CMU PIE database) facial image samples of each person for training, and the other ones are used for testing, the number of the homogeneous neighbors is set as i-1 respectively and the the number of the hetergeneous neighbors is equal to 10. The balance parameter β is 0.2 for all databases and the parameter for side information ϵ is 0.8. In practical application, all the parameters in BANMM can be optimized by cross-validation method or leave-one-out method [19] [20]. Fig.1 and Fig.2 demonstrate the growing trends of face classification rates corresponding to the increasing dimension of the feature. The best performances obtaind by different methods are given on Table 1.

It is clear that the proposed method BANMM is more effective than ANMM in extracting discriminant feature and reprsenting facial features over the varying lighting, facial expressions and pose. BANMM can achieve better performances than ANMM, especially in Yale database, the improvements are more than 5%

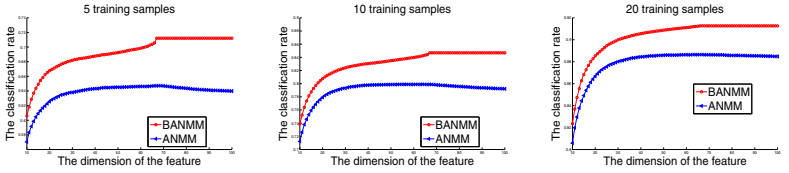


Fig. 2. The face recognition rates on CMU PIE database with 5, 10, 20 training samples for each person

Table 1. Face Recognition Rate on Three Datasets (%)

Method	Yale			YaleB			CMU PIE		
Training Number	3	4	5	5	10	20	5	10	20
PCA+LDA	60.70	67.19	74.13	65.08	78.26	85.94	57.18	75.31	84.52
ANMM	61.78	67.24	71.82	69.22	81.81	89.25	64.71	79.90	88.65
BANMM	**66.45**	**73.49**	**77.62**	**70.50**	**83.50**	**91.17**	**71.21**	**84.68**	**91.26**

in different training data sizes. In Fig.1 (d)-(f), one can see BANMM can reach a better performance by extracting less features than ANMM.

5 Conclusion

In this paper, a new supervised method for discriminative feature extraction called Balance Average Neighborhood Margin Maximization(BANMM) is proposed. As a new version of ANMM algorithm, the proposed method can preserve the locality of the original in the feature space and balance the weights of intra-class relationship and inter-class relationship in determining the projective map. Besides that, BANMM adopted a penalty term to remain the more discriminant information in the feature space than ANMM. The experimental results on three typical face database illustrate BANMM can derive a better feature space for face recognition than ANMM.

References

1. Jolliffe, I.: Principal Component Analysis. Springer, New York (1986)
2. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, New York (2001)
3. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans. on Pattern Analysis and Machine Intelligence 19(7), 711–720 (1997)
4. Chen, L.F., Liao, H.Y.M., Ko, M.T., Lin, J.C., Yu, G.J.: A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognition 33(10), 1713–1726 (2000)

5. Wang, F., Zhang, C.: Feature extraction by maximizing the average neighborhood margin. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
6. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3(1), 71–86 (1991)
7. Wang, X., Tang, X.: A unified framework for subspace face recognition. IEEE Trans on Pattern Analysis and Machine Intelligence 26(9), 1222–1228 (2004)
8. Wang, X., Tang, X.: Dual-space linear discriminant analysis for face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2004)
9. Liu, C., Wechsler, H.: Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. IEEE Trans on Image Processing 11(4), 467–476 (2002)
10. Li, H., Jiang, T., Zhang, K.: Efficient and robust feature extraction by maximum margin criterion. IEEE Trans. on Neural Networks 17(1), 157–165 (2006)
11. Lu, J., Plataniotis, K.N., Venetsanopoulos, A.N.: Face recognition using LDA-based algorithms. IEEE Trans. on Neural Networks 14(1), 195–200 (2003)
12. He, X., Niyogi, P.: Locality preserving projections (lpp). In: Advances in Neural Information Processing Systems (2003)
13. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.J.: Face recognition using laplacianfaces. IEEE Trans on Pattern Analysis and Machine Intelligence 27(3), 328–340 (2005)
14. Zhao, W., Chellappa, R., Krishnaswamy, A.: Discriminant analysis of principal components for face recognition. In: IEEE International Conference on Automatic Face and Gesture Recognition, pp. 336–341 (1998)
15. Qiu, X., Wu, L.: Face recognition by stepwise nonparametric margin maximum criterion. In: IEEE International Conference on Computer Vision (2005)
16. Yan, S., Xu, D., Zhang, B., Zhang, H.J.: Graph embedding: A general framework for dimensionality reduction. In: IEEE Conference on Computer Vision and Pattern Recognition (2005)
17. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning with application to clustering with side-information. In: Advances in Neural Information Processing Systems (2003)
18. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: Advances in Neural Information Processing Systems (2006)
19. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International joint Conference on artificial intelligence, pp. 1137–1145 (1995)
20. Devijver, P.A., Kittler, J.: Pattern recognition: A statistical approach. Prentice-Hall, London (1982)

The Relationship between the Newborn Rats' Hypoxic-Ischemic Brain Damage and Heart Beat Interval Information

Xiaomin Jiang[1], Hiroki Tamura[1], Koichi Tanno[1], Li Yang[2], Hiroshi Sameshima[2], and Tsuyomu Ikenoue[2]

[1] Faculty of Enginerring & Graduate School of Enginerring, University of Miyazaki, 1-1, Gakuen Kibanadai Nishi, Miyazaki, 889-2192, Japan
[2] Faculty of Medicine, University of Miyazaki, 5200, Kihara Kiyotake, Miyazaki, 889-1692, Japan
{tc10042@student,htamura@cc,tanno@cc}.miyazaki-u.ac.jp

Abstract. This research is aim to monitor the possibility of hypoxic-ischemic (abbr. HI) brain damage in newborn rats by studying and determining the newborns' heart rate/ R-R interval in turn to minimize the possibility of HI brain damage for human newborns during births. This research is based on the 20 newborn rats' heart rate/ R-R interval information during hypoxic insult. The data will be changed to the parameters Local Variation (Lv), Coefficient Variation (Cv), correlation coefficient (R^2), and then be analyzed using Multiple Linear Regression Analysis and Successive Multiple Linear Regression Analysis. This paper shows that it will be possible to predict the future development of HI brain damage in human fetus by using of heart rate/ R-R interval information.

Keywords: Hypoxic-ischemic brain damage (HI), heart rate/ R-R interval, Local Variation (Lv), Coefficient Variation (Cv), correlation coefficient (R^2), multiple linear regression analysis.

1 Background

Acute hypoxia-ischemia is an important factor in causing brain injury in term infants during labor [1]. According to the statistics, 2~4 of 1000 human newborns will occur hypoxic-ischemic (abbr. HI) brain damage, in which over 50% will lead to death or suffer in the long-term neurological abnormalities [2].

On the other hand, with the development of the engineering technology, many types of medical equipment make a great contribution to decrease the fetus mortality. Fetal heart rate (FHR) monitoring is well known as an effective method to assess fetal health [3]. However, it still has its limitations.

Previously, we showed that the possibility of predicting HI brain damage in newborn rats by analyzing the heart rate/ R-R interval information before and after hypoxic period [4].

In this study, we used a newborn rat model of HI brain damage [5], and investigated whether there is any significant associated with brain damage during hypoxic period.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 117–124, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Experiments

2.1 Data Collection

In this study, we used heart rate/R-R interval of the newborn rats. The animal experiment was approved by the University of Miyazaki Animal Care and Use Committee and was in accordance with the Japanese Physiological Society's guidelines for animal care. Rat pups were lightly anesthetized, and the left common carotid artery was legated. The wire electrodes were placed on the chest for the electrocardiogram (ECG). After 2 hours of recovery, they were exposed to hypoxia (oxygen 8%) for 150 minutes. Heart rate/R-R intervals during the hypoxic period were used for analyzing.

One week after HI insults, the rats were sacrificed by an intraperitioneal injection of a lethal dose of pentobarbital. The brains were removed, and embedded in paraffin. Each paraffin section was stain with hematoxylin-eosin (HE). The brain damage was evaluated under the microscope.

In this study, the 20 newborn rats were used, in which 11 rats showed no brain damage (Fig 1) and 9 rats showed brain damage (Fig 2).

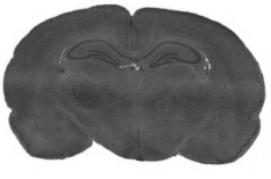


Fig. 1. Brain cross section of non-damage

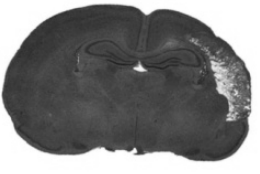


Fig. 2. Brain cross section of damage

2.2 Data Analyzing

Calculating the data of R-R intervals collected from the experiments into the engineering variations: *Lv*[6], *Cv* and R^2. R^2 is the correlation coefficient of *Lv* and *Cv*, which shows the relationship between *Lv* and *Cv* (Fig 3 and Fig 4) of 10 minutes R-R intervals. The value of R^2, which is larger than 0.8, shows that there is a close relationship between *Lv* and *Cv*. *Lv* is the local variation, which means the changes in the adjacent Inter Spike Interval (abbr. ISI). *Cv* is the coefficient variation, which means the changes in the total ISI. *Lv* and *Cv* are calculated with the following formulas:

$$Lv = \frac{1}{n-1}\sum_{i=1}^{n-1}\frac{3\left(T_i - T_{i-1}\right)^2}{\left(T_i + T_{i+1}\right)^2} \qquad Cv = \frac{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(T_i - \overline{T}\right)^2}}{\overline{T}}$$

T_i: *anyone of ISI* $\overline{T}$: *the average of ISI*

n: *the number of ISI*

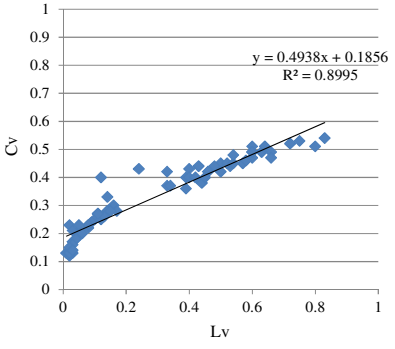


Fig. 3. Example of Correlation diagram of *Lv-Cv*: The brain damage was not generated in this newborn rat

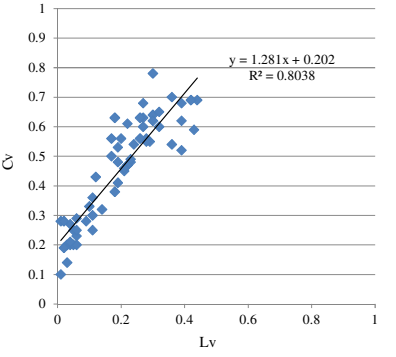


Fig. 4. Example of Correlation diagram of *Lv-Cv*: The brain damage was generated in this newborn rat

2.3 Multiple Linear Regression Analysis

Multiple Linear Regression Analysis (abbr. MLR) is a multivariate statistical technique for examining the linear correlations between two or more independent variables (abbr. IVs) and a single dependent variable (abbr. DV). It can be of the form "To what extent do IVs predict DV?" [7].

In this research, if the rat suffered in brain damage or not is the predicted DV. With the engineering variations calculated in section 2.2, define X_1, X_2 as the IVs of MLR and E_1 as the standard for DV. IVs are defined as the data determined in every 10 minutes during the 150 minutes' hypoxia.

$$X_1 = \max(Lv) - \min(Lv) + \max(Cv) - \min(Cv) + \max(R^2) - \min(R^2)$$
$$X_2 = \min(Lv) + \min(Cv) + \min(R^2)$$

X_1 is the range of the engineering variations Lv, Cv and R^2, which shows the total variable of the heart rate/ R-R interval information for each newborn rat in hypoxia. X_2 is the sum of the minim of the variations, which shows the total most stability situation of the rat. Predicted damage E_1 for each rat can be calculated using MLR. The coefficients a_0, a_1, a_2 are calculated during MLR.

$$E_1 = a_0 + a_1 X_1 + a_2 X_2$$

On the other hand, X_1 and X_2 can be resolved into six IVs: $X_3 \sim X_8$. At this time, $X_3 \sim X_8$ is the IVs of MLR and E_2 is the standard.

$$X_3 = \max(Lv) - \min(Lv) \quad X_4 = \max(Cv) - \min(Cv)$$
$$X_5 = \max(R^2) - \min(R^2)$$
$$X_6 = \min(Lv) \quad X_7 = \min(Cv) \quad X_8 = \min(R^2)$$

$X_3 \sim X_5$ shows the variable of each variation for every rat in hypoxia, while $X_6 \sim X_8$ shows the most stability situation of each variation of the rat. Predicted damage E_2 for

each rat can be calculated. The coefficients b_0, b_1, b_2, b_3, b_4, b_5, b_6 are calculated during MLR.

$$E_2 = b_0 + b_1 X_3 + b_2 X_4 + b_3 X_5 + b_4 X_6 + b_5 X_7 + b_6 X_8$$

2.4 Successive Multiple Linear Regression Analysis

In section 2.3, IVs is defined as the data determined in every 10 minutes during the 150 minutes' hypoxia, which is the unit in this research. Successive Multiple Linear Regression Analysis (abbr. SMLR) is on the base of MLR. It also uses the calculating way of MLR, however, the IVs is defined as the data of successive 50 minutes.

$$\begin{aligned} X_{1i} &= \max(Lv_i, Lv_{i-1}, Lv_{i-2}, Lv_{i-3}, Lv_{i-4}) - \min(Lv_i, \cdots, Lv_{i-4}) \\ &\quad + \max(Cv_i, \cdots, Cv_{i-4}) - \min(Cv_i, \cdots, Cv_{i-4}) \\ &\quad + \max(R_i^2, \cdots, R_{i-4}^2) - \min(R_i^2, \cdots, R_{i-4}^2) \\ X_{2i} &= \min(Lv_i, Lv_{i-1}, Lv_{i-2}, Lv_{i-3}, Lv_{i-4}) + \min(Cv_i, \cdots, Cv_{i-4}) + \min(R_i^2, \cdots R_{i-4}^2) \\ E_3 &= \max(a_o + a_1 X_{1i} + a_2 X_{2i}) \qquad (i: 5 \sim 15) \end{aligned}$$

The same of section 2.3, X_{1i} and X_{2i} can be resolved into as follows:

$$\begin{aligned} X_{3i} &= \max(Lv_i, \cdots, Lv_{i-4}) - \min(Lv_i, \cdots Lv_{i-4}) \\ X_{4i} &= \max(Cv_i, \cdots, Cv_{i-4}) - \min(Cv_i, \cdots, Cv_{i-4}) \\ X_{5i} &= \max(R_i^2, \cdots, R_{i-4}^2) - \min(R_i^2, \cdots, R_{i-4}^2) \\ X_{6i} &= \min(Lv_i, Lv_{i-1}, Lv_{i-2}, Lv_{i-3}, Lv_{i-4}) \\ X_{7i} &= \min(Cv_i, Cv_{i-1}, Cv_{i-2}, Cv_{i-3}, Cv_{i-4}) \\ X_{8i} &= \min(R_i^2, R_{i-1}^2, R_{i-2}^2, R_{i-3}^2, R_{i-4}^2) \\ E_4 &= \max(b_0 + b_1 X_{3i} + b_2 X_{4i} + b_3 X_{5i} + b_4 X_{6i} + b_5 X_{7i} + b_6 X_{8i}) \qquad (i: 5 \sim 15) \end{aligned}$$

The Fuzzy System has also been used on testing the same data in the experiment. We used the adaptive neuro-fuzzy inference system [8]. From the result, there was a good chance the relationship between HI brain damage and heart rate/ R-R interval existed when the 20 groups of data all used in the system. However, the test of leave one out cross validation in Fuzzy System was failed. Because of the large amount of arguments, over-fitting was considered to be the main reason for the failed test. It is thought that much less variables used in SMLR can avoid over-fitting.

3 Results

According to section 2.3 and 2.4, the results ($E_1 \sim E_4$) of MLR and SMLR are as Fig 5 to Fig 8. As it shows in the figures, x-ray is the actual brain damage results of the newborn rats used in the experiment. *0* means the rat did not suffer in the HI brain damage while *1* means the rat got HI brain damage at last. Y-ray is the predicted value of the MLR or SMLR. There is a border line in the figures to estimate the result.

If the predicted value smaller than the value of the border line, the newborn rat will be considered to be in the group of non-brain damage. However, if the predicted value bigger than the value of the border line, the rat will be considered as one of the group of brain damage.

Fig 5 shows the result of MLR with two IVs. Compared to the actual results, the rate of estimation is only 75%. However, the rate will rise to 85% if six IVs are used in MLR (Fig 6).

Fig 7 and Fig 8 show the results of SMLR. When there are only two IVs, the rate is 75% and if there are six IVs, the rate will be 85%. SMLR could be evaluated at intervals of 10 minutes. Therefore, it can be considered that the technique of SMLR is more effective and useful than the technique of MLR.

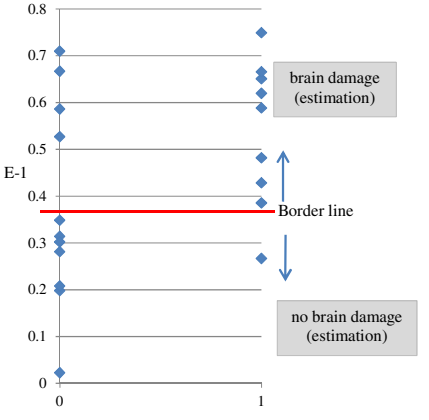


Fig. 5. The result of MLR with two independent variables (X_1 and X_2)

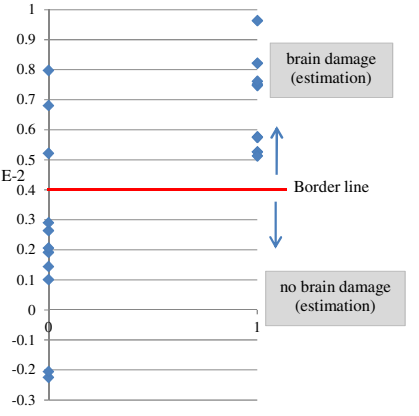


Fig. 6. The result of MLR with six independent variables ($X_3 \sim X_8$)

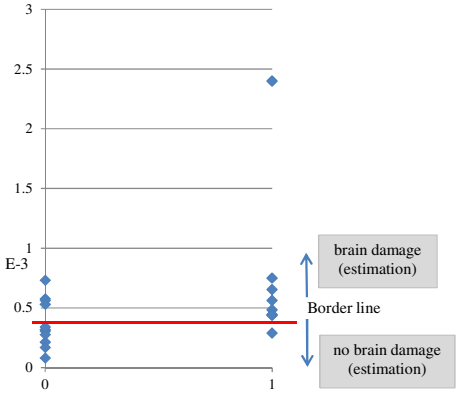


Fig. 7. The result of SMLR with two independent variables (X_1 and X_2)

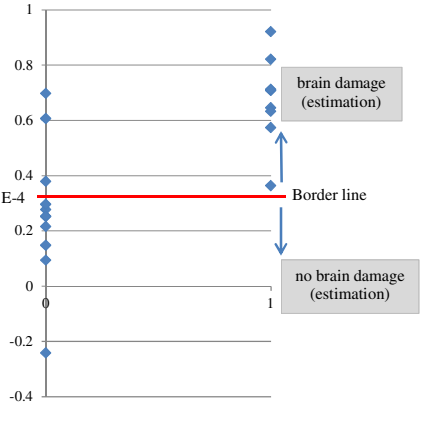


Fig. 8. The result of SMLR with six independent variables ($X_3 \sim X_8$)

4 Conclusions

This research is using the data of heart rate/R-R interval of the rat newborns with hypoxia-ischemia, to determine the possibility of HI brain damage for human newborns. As it shown above, there is 85% possibility to predict the damage.

However, how to predict the damage is also a problem. Because of the highest possibilities of SMLR with six independent variables in this research, the predicted damage E_i for each newborn rat is calculated by SMLR with six variables. Define E_d as the average value of E_i of the group of brain damages and E_n as the average value of E_i of the group of non-brain damage.

Fig 9 shows the changes of E_d, E_n from the time of 50 minutes to 120 minutes. The value of E_n is much lower than E_d, and keeps below 0 through the experiments. As a result, it may be distinct as non-brain damage happened if the value of predicted damage keeps negative. On the other hand, the value of E_d is growing after the 90th minute and reaches to the highest in the time point of 120th minute. However, the time point of 90th minute and 120th minute may too late to predict the brain damage in practical application, which will be optimized in the future researches.

Fig 10 shows the change of ISI for each newborn rat. Define the difference between the ISI at the beginning and the end of the hypoxic period as the change of ISI. X-ray is the value of E_4 calculated in section 2.4, while Y-ray is the change of ISI. The change of non-brain damage spreads in the whole area when E_4 is positive, and the change of brain damage gathers in a small area with the value of E_4 larger than the border line and the value of ISI change smaller than -9. From Fig 10, there is 95% possibility to predict the damage.

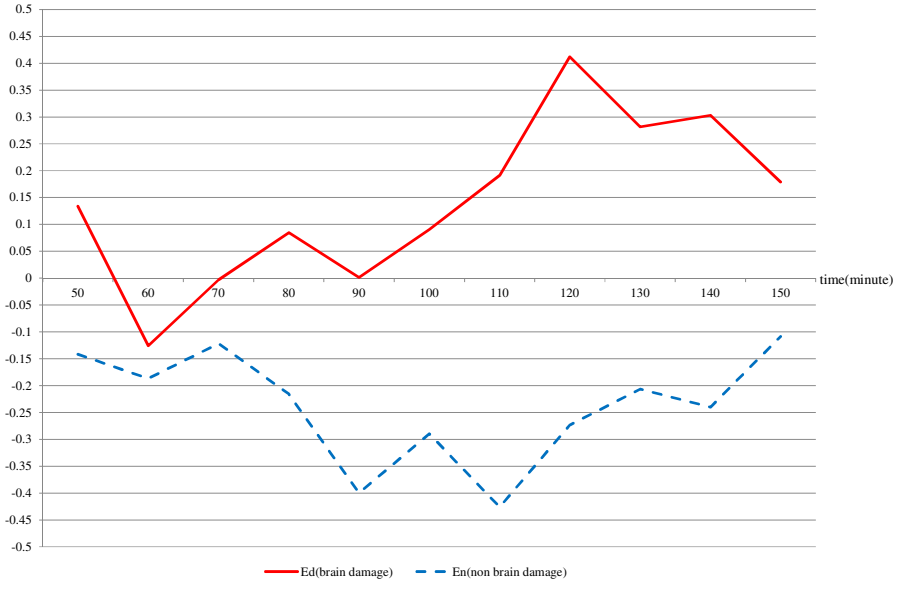


Fig. 9. The change of E_d, E_n

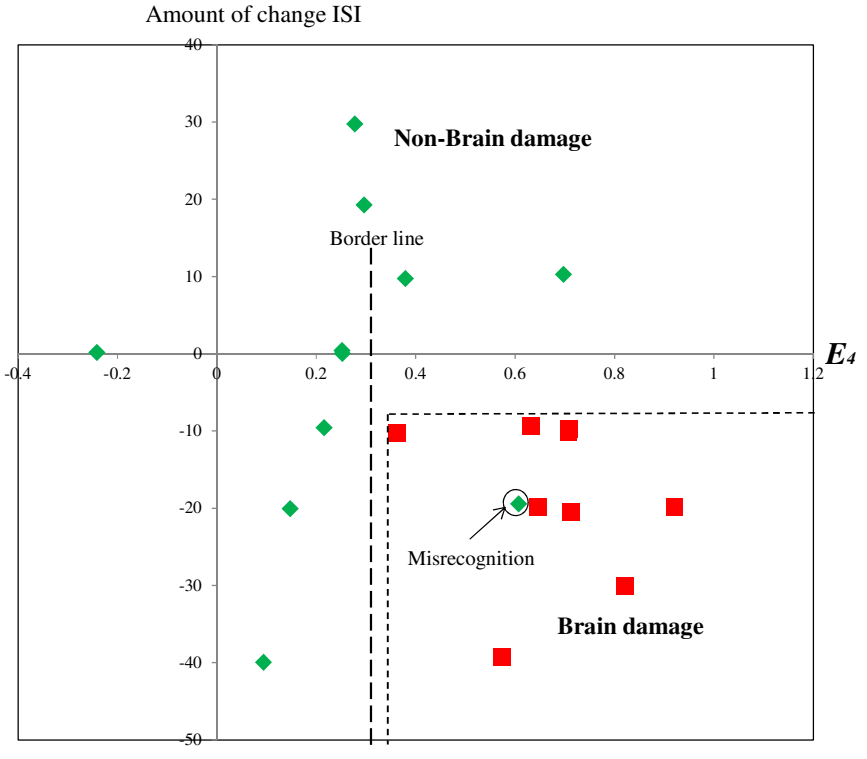


Fig. 10. The change of ISI and *E4*

In conclusion, distinct the HI brain damage for human newborns during birth is possible. But how to distinct is still a very important study for the next researches. More newborn rats in the experiment, more IVs in the analysis and more relationship between the standard *E* and the time point will be the main points in the next research.

References

1. Hill, A.: Current concepts of hypoxic-ischemic cerebral injury in the term newborn. Pediatr. Neurol. 7, 317–325 (1991)
2. Volpe, J.J.: Neurology of the Newborn. W.B.Saunders, Philadelphia (2000)
3. Phelan, J.P., Kim, J.O.: Fetal heart rate observations in the braindamaged infant. Semin Perinatol 24, 221–229 (2000)
4. Tamura, H., Yang, L., Tanno, K., Murao, K., Sameshima, H., Ikenoue, T.: A Study on The Distinction Method of The Newborn Rat Brain Damage using Heart Beat Interval Information. Japanese Society for Medical and Biological Engineering (JSBME) 47(6), 618–622 (2009)

5. Ota, A., Ikeda, T., Ikenoue, T., Toshimori, K.: Sequence of neuronal responses assessed by immunohistochemistry in the newborn rat brain after hypoxia-ischemia. Am. J. Obstet Gynecol. 177(3), 519–526 (1997)
6. Shinomoto, S., Shima, K., Tanji, J.: Differences in spiking patterns among cortical neuron. Neural Computation 15, 2823–2842 (2003)
7. Shinomoto, S.: Prediction and simulation. Iwanami-Shoten Publishers (2002)
8. Jang, J.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. 23(3), 665–685 (1993)

A Robust Approach for Multivariate Binary Vectors Clustering and Feature Selection

Mohamed Al Mashrgy[1], Nizar Bouguila[1], and Khalid Daoudi[2]

[1] Concordia University, QC, Cannada
m_almash@encs.concordia.ca, bouguila@ciise.concordia.ca
[2] INRIA Bordeaux Sud Ouest, France
khalid.daoudi@inria.fr

Abstract. Given a set of binary vectors drawn from a finite multiple Bernoulli mixture model, an important problem is to determine which vectors are outliers and which features are relevant. The goal of this paper is to propose a model for binary vectors clustering that accommodates outliers and allows simultaneously the incorporation of a feature selection methodology into the clustering process. We derive an EM algorithm to fit the proposed model. Through simulation studies and a set of experiments involving handwritten digit recognition and visual scenes categorization, we demonstrate the usefulness and effectiveness of our method.

Keywords: Binary vectors, Bernoulli, outliers, feature selection.

1 Introduction

The problem of clustering, broadly stated, is to group a set of objects into homogenous categories. This problem has attracted much attention from different disciplines as an important step in many applications [1]. Finite mixture models have been widely used in pattern recognition and elsewhere as a convenient formal approach to clustering and as a first choice off the shelf for the practitioner. The main driving force behind this interest in finite mixture models is their flexibility and strong theoretical foundation. The majority of mixture-based approaches have been based on the Gaussian distribution. Recent researches have shown, however, that this choice is not appropriate in general especially when we deal with discrete data and in particular binary vectors [2]. The modeling of binary data is interesting at the experimental level and also at a deeper theoretical level. Indeed, this kind of data is naturally and widely generated by various pattern recognition and data mining applications. For instance, several image processing and pattern recognition applications involve the conversion of grey level or color images into binary images using filtering techniques. A given document (or image) can be represented by a binary vector where each binary entry describes the absence or presence of a given keyword (or visual word) in the document (or image) [3]. An important problem is then the development of statistical approaches to model and cluster such binary data.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 125–132, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Several previous researches have addressed the problem of binary vectors classification and clustering. For example, a likelihood ratio classification method based on Markov chain and Markov mesh assumption has been proposed in [4]. A kernel-based method for multivariate binary vectors discrimination has been proposed in [5]. A fuzzy sets-based clustering approach has been proposed in [6] and applied for medical diagnosis. An evaluation of five discrimination approaches for binary data has been proposed in [7]. A multiple cause model for the unsupervised learning of binary data has been proposed in [8]. Recently, we have tackled the problem of unsupervised binary feature selection by proposing a statistical framework based on finite multivariate Bernoulli mixture models which has been applied successfully to several data mining and multimedia processing tasks [2,3,9]. In this paper, we go a step further by tackling simultaneously, with clustering and feature selection, the challenging problem of outlier detection. We are mainly motivated by the fact that learning algorithms should provide accurate, efficient and robust approaches for prediction and classification which can be compromised by the presence of outliers as shown in several research works (see, for instance, [1,10]). To the best of our knowledge the well-known data clustering algorithms offer no solution to the combination of feature selection and outlier rejection in the case of binary data.

The rest of this paper is organized as follows. First, we present our model and an approach to learn it in the next section. This is followed by some experimental results in Section 3 where we give results on a benchmark problem in pattern recognition namely the classification of handwritten digits and in a second problem which concerns visual scenes categorization. Finally, we end the article with some conclusions as well as future issues for research.

2 A Model for Simultaneous Clustering, Feature Selection and Outliers Rejection

In this section we first describe our statistical framework for simultaneous clustering, feature selection and outliers rejection using finite multivariate Bernoulli mixture models. An approach to learn the proposed statistical model is then introduced and a complete EM-based learning algorithm is proposed.

2.1 The Model

Let $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\} \in \{0,1\}^D$ be a set of D-dimensional binary vectors. In a typical model-based cluster analysis, the goal is to find a value $M < N$ such that the vectors are well modeled by a multivariate Bernoulli mixture with M components:

$$p(\boldsymbol{X}_n|\Theta_M) = \sum_{j=1}^{M} p_j p(\boldsymbol{X}_n|\boldsymbol{\pi}_j) = \sum_{j=1}^{M} p_j \prod_{d=1}^{D} \pi_{jd}^{X_{nd}} (1-\pi_{jd})^{1-X_{nd}} \quad (1)$$

where $\Theta_M = \{\{\boldsymbol{\pi}_j\}, \boldsymbol{P}\}$ is the set of parameters defining the mixture model, $\boldsymbol{\pi}_j = (\pi_{j1}, \ldots, \pi_{jD})$ and $\boldsymbol{P} = (p_1, \ldots, p_M)$ is the mixing parameters vector,

$0 \leq p_j \leq 1$, $\sum_{j=1}^{M} p_j = 1$. It is noteworthy that the previous model assumes actually that all the binary features have the same importance. It is well-known, however, that in general only a small part of features may allow the differentiation of the different present clusters. This is especially true when the dimensionality increases and in this case the so-called curse of dimensionality becomes problematic in part because of the sparseness of data in higher dimensions. In this context many of the features may be irrelevant and will just introduce noise and then compromise the uncovering of the clustering structure [11]. A major advance in feature selection was made in [12] where the problem was defined within finite Gaussian mixtures. In [2,3], we adopted the approach in [12] to tackle the problem of unsupervised feature selection in the case of binary vectors by proposing the following model

$$p(\boldsymbol{X}_n|\Theta) = \sum_{j=1}^{M} p_j \prod_{d=1}^{D} \left[\rho_d \pi_{jd}^{X_{nd}} (1-\pi_{jd})^{1-X_{nd}} + (1-\rho_d) \lambda_d^{X_{nd}} (1-\lambda_d)^{1-X_{nd}} \right] \quad (2)$$

where $\Theta = \{\Theta_M, \{\rho_d\}, \boldsymbol{\Lambda}\}$, $\boldsymbol{\Lambda} = (\lambda_1, \ldots, \lambda_D)$ are the parameters of a multivariate Bernoulli distribution considered as a common background model to explain irrelevant features, and $\rho_d = p(\phi_d = 1)$ is the probability that feature d is relevant such that ϕ_d is a missing value equal to 1 if feature d is irrelevant and equal to 0, otherwise. Feature selection is important not only because it allows the determination of relevant modeling features but also because it provides understandable, scalable and more accurate models that prevent data under- or over-fitting. Unfortunately, the modeling capabilities in general and the feature selection process in particular can be negatively affected by the presence of outliers. Indeed, a common problem in machine learning and data mining is to determine which vectors are outliers when the data statistical model is known. Removing these outliers will normally enhance generalization performance and interpretability of the results. Moreover, it is well-known that the success of many applications usually depends on the detection of potential outliers which can be viewed as unusual data that are not consistent with most observations. Classic works on outlier rejection have considered being an outlier as a binary property (i.e. either the vector in the data set is an outlier or not). In this paper, however, we argue that it is more appropriate to affect to each vector a degree (i.e. a probability) of being an outlier or not as it has been shown also in some previous works [10]. In particular, we define a cluster independent outlier vector to be one that can not be represented by any of the mixture's components and then associated with a uniform distribution having a weight equal to p_{M+1} indicating the degree of outlier-ness. This can be formalized as follow

$$p(\boldsymbol{X}_n|\Theta) = \sum_{j=1}^{M} p_j \prod_{d=1}^{D} [\rho_d \pi_{jd}^{X_{nd}} (1-\pi_{jd})^{1-X_{nd}} + (1-\rho_d) \lambda_d^{X_{nd}} (1-\lambda_d)^{1-X_{nd}}] + p_{M+1} U(\boldsymbol{X}_n) \quad (3)$$

where $p_{M+1} = 1 - \sum_{j=1}^{M} p_j$ is the probability that $\boldsymbol{X}_n$ was not generated by the central mixture model and $U(\boldsymbol{X}_n)$ is a uniform distribution common for all data

to model isolated vectors which are not in any of the M clusters and which show significantly less differentiation among clusters. Notice that when $p_{M+1} = 0$ the outlier component is removed and the previous equation is reduced to Eq. 2.

2.2 Model Learning

The EM algorithm, that we use for our model learning, has been shown to be a reliable framework to achieve accurate estimation of mixture models. Two main approaches may be considered within the EM framework namely maximum likelihood (ML) estimation and maximum a posteriori (MAP) estimation. Here, we use MAP estimation since it has been shown to provide accurate estimates in the case of binary vectors [2,3]:

$$\hat{\Theta} = \arg\max_{\Theta} \{\log p(\mathcal{X}|\Theta) + \log p(\Theta)\} \tag{4}$$

where $\log p(\mathcal{X}|\Theta) = \log \prod_{i=1}^{N} p(\boldsymbol{X}_n|\Theta)$ is our model's loglikelihood function and $p(\Theta)$ is the prior distribution and is taken as the product of the priors of the different model's parameters. Following [2,3], we use a Dirichlet prior with parameters $(\eta_1, \ldots, \eta_{M+1})$ for the mixing parameters $\{p_j\}$ and Beta priors for the multivariate Bernoulli distribution parameters $\{\pi_{jd}\}$. Having these priors in hand, the maximization in Eq. 4 gives us the following

$$p_j = \frac{\sum_{n=1}^{N} p(j|\boldsymbol{X}_n) + (\eta_j - 1)}{N + M(\eta_j - 1)} \quad j = 1, \ldots, M+1 \tag{5}$$

where

$$p(j|\boldsymbol{X}_n) = \begin{cases} \frac{p_j \prod_{d=1}^{D} (\rho_d p_{jd}(X_{nd}) + (1-\rho_d) p(X_{nd}))}{\sum_{j=1}^{M} (p_j \prod_{d=1}^{D} (\rho_d p_{jd}(X_{nd}) + (1-\rho_d) p(X_{nd}))) + p_{M+1} U(\boldsymbol{X}_n)} & \text{if } j = 1, \ldots, M \\ \frac{p_{M+1} U(\boldsymbol{X}_n)}{\sum_{j=1}^{M} (p_j \prod_{d=1}^{D} (\rho_d p_{jd}(X_{nd}) + (1-\rho_d) p(X_{nd}))) + p_{M+1} U(\boldsymbol{X}_n)} & \text{if } j = M+1 \end{cases} \tag{6}$$

where $p_{jd}(X_{nd}) = \pi_{jd}^{X_{nd}}(1-\pi_{jd})^{1-X_{nd}}$ and $p(X_{nd}) = \lambda_d^{X_{nd}}(1-\lambda_d)^{1-X_{nd}}$. $p(j|\boldsymbol{X}_n)$ is the posterior probability that a vector $\boldsymbol{X}_n$ will be considered as an inlier and then assigned to a cluster $j, j = 1, \ldots, M$ or as an outlier and then affected to cluster $M+1$. Details about the estimation of the other model parameters namely π_{jd}, λ_d, and ρ_d can be found in [2,3]. The determination of the optimal number of clusters is based on the Bayesian information criterion (BIC) [13]. Finally, our complete algorithm can be summarized as follows

Algorithm

For each candidate value of M:

1. Set $\rho_d \leftarrow 0.5$, $d = 1, \ldots, D$, $j = 1, \ldots, M$ and initialization of the rest of parameters using the K-Means algorithm by considering that $M+1$ clusters.
2. Iterate the two following steps until convergence:
   (a) E-Step: Update $p(j|\boldsymbol{X}_n)$ using Eq. 6.
   (b) M-Step: Update the p_j using Eq. 5 (the η_j are set to 2), and π_{jd}, λ_d and ρ_d as done in [2].
3. Calculate the associated BIC.
4. Select the optimal model that yields the highest BIC.

3 Experimental Results

In this section, we validate our approach via two applications. The first one concerns handwritten digit recognition and the second one tackles visual scenes categorization.

3.1 Handwritten Digit Recognition

In this first application which concerns the challenging problem of handwritten digit recognition (see, for instance, [14]), we use a well-known handwritten digit recognition database namely the UCI data set [15]. The UCI database contains 5620 objects. The repartition of the different classes is given in table 1. The original images are processed to extract normalized bitmaps of handwritten digits. Each normalized bitmap includes a 32×32 matrix (each image is represented then by 1024-dimensional binary vector) in which each element indicates one pixel with value of white or black. Figure 1 shows an example of the normalized bitmaps. For our experiments we add also 50 additional binary images (see Fig. 2), which are taken from the MPEG-7 shape silhouette database [16] and do not contain real digits, to the UCI data set. These additional images are considered as the outliers. Evaluation results by considering different scenarios: recognition without feature selection and without outliers rejection (Rec), recognition with feature selection and without outlier rejection (RecFs), recognition without feature selection and with outliers rejection (RecOr), and recognition with feature selection and outlier rejection (RecFsOr) are summarized in table 2. It is noteworthy that we were able to find the exact number of clusters



Fig. 1. Example of normalized bitmaps

Table 1. Repartition of the different classes

class	0	1	2	3	4	5	6	7	8	9
Number of objects	554	571	557	572	568	558	558	566	554	562



Fig. 2. Examples of the 50 images taken from the MPEG-7 shape silhouette database and added as outliers

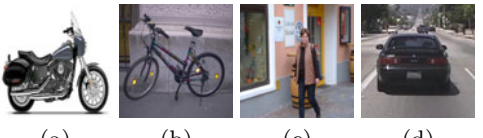
Table 2. Error rates for the UCI data set by considering different scenarios

Rec	RecFs	RecOr	RecFsOr
14.37%	10.21%	9.30%	5.10%

only when we have rejected the outliers. According to the results in table 2 it is clear that feature selection improves the recognition performance especially when combined with outliers rejection.

3.2 Visual Scenes Categorization

Here, we consider the problem of visual scenes categorization by considering the challenging PASCAL 2005 corpus which has 1578 labeled images grouped into 4 categories (motorbikes, bicycles, people and cars) as shown in Fig. 3 [17]. In particular, we use the approach that we have previously proposed in [3] which consists on representing visual scenes as binary vectors and which can be summarized as follows. First, interest points are detected on images using the difference-of-Gaussians point detector [18]. Then, we use PCA-SIFT descriptor [19] which allows the description of each interest point as a 36-dimensional vector. From the considered database, images are taken, randomly, to construct the visual vocabulary. Moreover, extracted SIFT vectors are clustered using the K-Means algorithm providing 5000 visual-words. Each image is then represented by a 5000-dimensional binary vector describing the presence or the absence of a set of visual words, provided from the constructed visual vocabulary. We add 60 outlier images from different sources to the PASCAL data set. In order to investigate the performance of our learning approach, we ran the clustering experiment 20 times. Over these 20 runs, the clustering algorithm successfully selected the exact number of clusters, which is equal to 4, 11 times and 5 times with and without feature weighting, respectively, when outliers were taken into account. Without outliers rejection, we were unable to find the exact number of clusters. Table 3 summarizes the results and it is clear again that the consideration of both feature selection and outliers rejection improves the results.



(a) (b) (c) (d)

Fig. 3. Example of images from the PASCAL 2005 corpus. (a) motorbikes (b) bicycles (c) people (d) cars.

Table 3. Error rates for the visual scenes categorization problem by considering different scenarios

Cat	CatFs	CatOr	CatFsOr
34.02%	32.43%	29.10%	27.80%

4 Conclusion

In this paper we have presented a well motivated approach for simultaneous binary vectors clustering and feature selection in the presence of outliers. Our model can be viewed as a way to robustify the unsupervised feature selection approach previously proposed in [2,3], to learn the right meaning from the right observations (i.e inliers). Experimental results that address issues arising from two applications namely handwritten digit recognition and visual scenes categorization have been presented. The main goal in this paper was actually the rejection of the outliers. Some works, however, have shown that these outliers may provide useful information and an expected knowledge, such as in electronic commerce and credit card fraud, as argued in [20] (i.e. "One person's noise is another person's signal" [20]). Thus a possible future application of our work could be of the extraction of useful knowledge from the detected outliers for applications like intrusion detection [21].

Acknowledgment. The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proc. of KDD, pp. 226–231 (1996)
2. Bouguila, N., Daoudi, K.: A Statistical Approach for Binary Vectors Modeling and Clustering. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 184–195. Springer, Heidelberg (2009)
3. Bouguila, N., Daoudi, K.: Learning Concepts from Visual Scenes Using a Binary Probabilistic Model. In: Proc. of IEEE International Workshop on Multimedia Signal Processing (MMSP), pp. 1–5 (October 2009)
4. Abend, K., Harley, T.J., Kanal, L.N.: Classification of Binary Random Patterns. IEEE Transactions on Information Theory 11(4), 538–544 (1965)
5. Aitchison, J., Aitken, C.G.G.: Multivariate Binary Discrimination by the Kernel Method. Biometrika 63(3), 413–420 (1976)
6. Bezdek, J.C.: Feature Selection for Binary Data: Medical Diagnosis with Fuzzy Sets. In: Proc. of the National Computer Conference and Exposition, New York, NY, USA, pp. 1057–1068 (1976)
7. Moore II, D.H.: Evaluation of Five Discrimination Procedures for Binary Variables. Journal of the American Statistical Association 68(342), 399–404 (1973)
8. Saund, E.: Unsupervised Learning of Mixtures of Multiple Causes in Binary Data. In: Advances in Neural Information Processing Systems (NIPS), pp. 27–34 (1993)

9. Bouguila, N.: On multivariate binary data clustering and feature weighting. Computational Statistics & Data Analysis 54(1), 120–134 (2010)
10. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: LOF: Identifying Density-Based Local Outliers. In: Proc. of the ACM SIGMOD International Conference on Management of Data (MOD), pp. 93–104 (2000)
11. Boutemedjet, S., Ziou, D., Bouguila, N.: Unsupervised Feature Selection for Accurate Recommendation of High-Dimensional Image Data. In: Advances in Neural Information Processing Systems (NIPS), pp. 177–184 (2007)
12. Law, M.H.C., Figueiredo, M.A.T., Jain, A.K.: Simultaneous Feature Selection and Clustering Using Mixture Models. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), 1154–1166 (2004)
13. Schwarz, G.: Estimating the Dimension of a Model. Annals of Statistics 16, 461–464 (1978)
14. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: Proc. of ICML, pp. 148–156 (1996)
15. Blake, C.L., Merz, C.J.: Repository of Machine Learning Databases. University of California, Irvine, Dept. of Information and Computer Sciences (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
16. Jeannin, S., Bober, M.: Description of core experiments for MPEG-7 motion/shape. Technical Report ISO/IEC JTC 1/SC 29/WG 11 MPEG99/N2690, MPEG-7 Visual Group, Seoul (March 1999)
17. Everingham, M., Zisserman, A., Williams, C.K.I., Van Gool, L., Allan, M., Bishop, C.M., Chapelle, O., Dalal, N., Deselaers, T., Dorkó, G., Duffner, S., Eichhorn, J., Farquhar, J.D.R., Fritz, M., Garcia, C., Griffiths, T., Jurie, F., Keysers, D., Koskela, M., Laaksonen, J., Larlus, D., Leibe, B., Meng, H., Ney, H., Schiele, B., Schmid, C., Seemann, E., Shawe-Taylor, J., Storkey, A.J., Szedmak, S., Triggs, B., Ulusoy, I., Viitaniemi, V., Zhang, J.: The 2005 PASCAL Visual Object Classes Challenge. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 117–176. Springer, Heidelberg (2006)
18. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
19. Ke, Y., Sukthankar, R.: PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In: Proc. of IEEE CVPR, pp. 506–513 (2004)
20. Knorr, E.M., Ng, R.T.: Algorithms for Mining Distance-Based Outliers in Large Datasets. In: Proc. of 24rd International Conference on Very Large Data Bases (VLDB), pp. 392–403 (1998)
21. Durst, R., Champion, T., Witten, B., Miller, E., Spagnuolo, L.: Testing and Evaluating Computer Intrusion Detection Systems. Commun. ACM 42, 53–61 (1999)

The Self-Organizing Map Tree (SOMT) for Nonlinear Data Causality Prediction

Younjin Chung and Masahiro Takatsuka

ViSLAB, The School of IT, The University of Sydney, NSW 2006 Australia

Abstract. This paper presents an associated visualization model for the nonlinear and multivariate ecological data prediction processes. Estimating impacts of changes in environmental conditions on biological entities is one of the required ecological data analyses. For the causality analysis, it is desirable to explain complex relationships between influential environmental data and responsive biological data through the process of ecological data predictions. The proposed Self-Organizing Map Tree utilizes Self-Organizing Maps as nodes of a tree to make association among different ecological domain data and to observe the prediction processes. Nonlinear data relationships and possible prediction outcomes are inspected through the processes of the SOMT that shows a good predictability of the target output for the given inputs.

Keywords: Nonlinear Data Relationships and Prediction Processes, Artificial Neural Network, Information Visualization, Self-Organizing Map.

1 Introduction

Data analyses to discover unknown and potentially useful information often deal with highly complex, nonlinear and multivariate data. In ecology, biological data are influenced by interactions of various types of environmental factors. Understanding the nature and the interactions of such ecological data has become increasingly significant in order to make better decisions in solving environmental problems [13]. Many methods and processes have been developed to understand complex relationships of ecological data and to predict possible environmental impacts on biological quality. Traditional statistical or ordination methods have yielded to novel approaches using Artificial Neural Networks (ANNs) for nonlinear ecological data analyses over a decade [6,10].

The research into ANNs becomes imperative when focusing on nonlinear data analyses. Different ANNs have been applied for different purposes. Unsupervised ANNs such as Self-Organizing Map (SOM) have been used for identifying data relationships while supervised ANNs such as Backpropagation Network (BPN) have been typically used for data predictions [4,13]. However, the informatoin obtained by each different type of networks is quite independent; they cannot be used in association with each other. The challenge for the mutual data analyses is to develop an interactive method, which allows analysts to carry out effective predictions with extracting causal relationships between complex and nonlinear

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 133–142, 2011.
© Springer-Verlag Berlin Heidelberg 2011

data. Providing an effective visualization for the method also helps people inspect different levels of information more efficiently.

SOM has contributed to the ecological data relationship analysis with its data patterning capability and visualization techniques [7], and BPN has been suggested for the prediction analysis [14]. However, the causalites among ecological data cannot be easily explained with the predciton of BPN since it neither interacts with SOM nor explains any data relationships. Besides, BPN produces only one output for an input through its typical prediction process. This process cannot generate other possibilities of predicting such as many outputs for an input and one target output by many inputs for ecological data as explained in Section 3. In order to address these issues, our SOM Tree (SOMT) uses SOMs as nodes of a tree for capturing correlations among multiple data types for nonlinear data predictions. The SOMT supports the propounded ecological data predictions against the BPN's typical prediction and the inspection of data relationships through the prediction processes.

The following section presents an overview of nonlinear ecological data analyses using ANNs, and the issues raised are stated in Section 3. The proposed SOMT with a novel prediction procedure is introduced in Section 4. Experimental results are given in Section 5, followed by the conclusion in Section 6.

2 Background

2.1 Nonlinear Data Relationship Analysis Using SOM

Discovering complex and nonlinear data relationships has been the primary data analysis in ecology [1,13]. Many ecologists have evaluated the effectiveness of ANNs through empirical comparisons against other conventional methods. Their emphasis on self-selection, ordination and classification with efficient visualizations for the relationship analysis positioned SOM into the centre of their approaches [1,6,10].

Since Chon et al. [2] utilized a SOM to explore biological data space in 1996, SOM has been increasingly applied to ecological research. It represented biological data of similar patterns, and the intra-relationships between biological variables were observed through pattern recognition. A multi-level SOM was also used by Tran et al. [16] to provide different views of the same environmental data at different scales. According to the studies, either biological or environmental domain data have been analyzed for their intra-relationships as most of analysis methods including SOM are able to deal with only a given single data set.

A few methods have been proposed using SOMs in order to study interrelationships between biological and environmental domain data of a given ecosystem. Park et al. [14] fed environmental variables to a SOM, which was previously trained with biological variables. The mean values of environmental variables were projected onto the SOM neurons. This approach has influenced the works of [3] and [13]. However, the method does not yield clear patterns of environmental data; it is not relevant for subsequent quantitative statistical analysis of the

relationships. Another approach was to train a SOM with a set of combined biological and environmental data to analyze these disparate data simultaneously [12]. This method seems to fit better for investigating the inter-relationships since each data attribute shows relative patterns on the SOM.

2.2 Nonlinear Data Prediction Analysis Using BPN

Predicting changes of biological data (profiles) according to environmental conditions has been the major concern in ecological sciences. The degree of environmental disturbances can be assessed with the biological profile information [14]. Among supervised ANNs, BPN has been the most used nonlinear predictor in estimating an output object for a given input object [10].

A BPN was used by Park et al. [14] in order to predict biological abundance according to a set of environmental conditions of an aquatic ecoregion. It was trained with a set of physical data, which is a type of environmental data, as the input for the desired output of biological data. After learning the relationships between the input and the output data, the target output for an input was predicted through its trained hidden layer. The result in their experiment using the BPN showed the high predictability with the accuracy rate of 0.91 for the trained data and 0.61 for the test data. However, the relationships between data cannot be described through the hidden processing layer, and difficulties are identified in explaining possible causalities among data. Although explaining data relationships might not be sufficient in terms of causality, it is fundamental in assessing environmental impacts on biological quality.

3 Issues of Nonlinear Data Prediction Process

It is ideal if ecological data can be sampled and analyzed within a pristine condition for all regions. However, most regions have been modified by human activities, and different regions have different ecological features. With this phenomenon, biological quality can be measured diversely at regional scales by alterations of various environmental factors [3].

BPN processes 'one-to-one' prediction, where only one output is predicted for an input. With the prediction process, there could be questions for such inconsistent ecological data as mentioned above, and two prediction cases are considered in this study. They are: 'one-to-many' case of predicting many biological responses for one type of environmental data (e.g. physical conditions only) and 'many-to-one' case of predicting the target biological profile by many types of environmental data (e.g. physical, chemical and land use conditions).

Furthermore, unlike SOM[1], BPN takes a set of environmental variables as the input for the desired biological output. This approach describes what an input and an output are but does not explain any relationships between the input and the output data since it does not allow observing the process of the hidden layer

[1] SOM takes a set of input data and the output is the patterns of the input space.

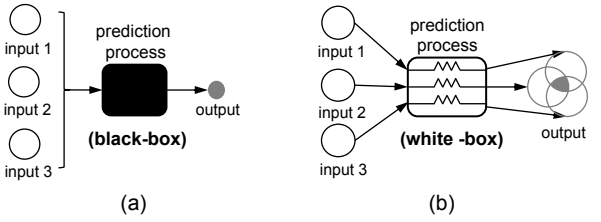


Fig. 1. Conceptual models of prediction process. (a) *'black-box'* model takes different inputs all together as an input for the target output. The process cannot be observed. (b) *'white-box'* model takes each input separately for each output and the target output is the common output by all inputs. The process can be observed.

(Figure 1(a)). Such a black boxed prediction process makes it difficult to conduct the causality analysis for assisting management decision makings.

Figure 1(b) describes a 'white-box' model comparing with the 'black-box' approach of BPN for the prediction processes. The 'white-box' approach is proposed to address the issues of inspecting data relationships through the prediction processes and supporting the two prediction hypotheses ('one-to-many' and 'many-to-one' cases) for nonlinear and multivariate ecological data.

4 The Self-Organizing Map Tree (SOMT)

4.1 Structure of the SOMT and the Prediction Processes

Based on the Kohonen's Self-Organizing Feature Map [8,9] and its great capability of exploring nonlinear ecological data relationships as described in Section 2.1, a new prediction method is proposed using the SOMs. The SOMs are organized in a tree structure named SOM Tree (SOMT) for the prediction analysis. In this study, we implemented our SOMT as a binary tree; however, it can take a general tree data structure. The SOMT is designed not to classify a single set of a data type into known categories such as a classification tree of Support Vector Machines (SVMs) [11]. It is designed to branch two correlative sets of different data types out to two child nodes from their parent node. Hence, the SOMT becomes a tree for correlating multiple data sets as depicted in Figure 2. This is different from previously reported Tree-SOM (TSOM), which organizes hierarchical SOMs to handle a single domain data set at different levels of details [15]. In the SOMT, each SOM at the external (child) node of the tree is trained with a separate domain data set of sample data. A SOM at the internal (parent) node associates the two external SOMs and captures the pair-wise relationships of the separate domains. The aim of the SOMT structure is to preserve information of data relationships for data predictions. Each external SOMs keep structural information of each domain data while the internal SOM explains the inter-relationships between the two different domain data by collating the contribution of each component.

Let environmental data vector as $E_n = [e_{n1}\ e_{n2}\ ...\ e_{ne}] \in R^e$ and biological data vector as $B_n = [b_{n1}\ b_{n2}\ ...\ b_{nb}] \in R^b$ of sampling site $S_n(n = 0, 1,..., s$: where s is the number of sites). Two external SOMs are trained with an environmental data set of E_n (ENV SOM) and a biological data set of B_n (BIO SOM) respectively. A combined data set can be created from these two data sets and $C_n = [c_{n1}\ c_{n2}\ ...\ c_{n(e+b)}] \in R^{e+b}$ denotes combined data vector of E_n and B_n. This combined data set is used for training the internal SOM (ENV-BIO SOM). ENV SOM is hence associated with BIO SOM through ENV-BIO SOM.

With the SOMT, various hypotheses can be generated and the following two prediction hypothesis generation processes are considered in this study:

1. 'One-to-many' prediction: starting with a neuron on one side external SOM and traverse the internal SOM of the tree to infer all possible corresponding neurons on the other side external SOM,
2. 'Many-to-one' prediction: starting with each neuron on the multiple external SOMs of one side and traverse each internal SOM of the tree to reach the common corresponding neuron(s) on the other side external SOM.

The prediction processes can be observed by highlighting its active neurons simultaneously on each SOM. Figure 2(a) presents a visual flow of the prediction processes for ecological data. Once the Best Matching Unit (BMU) for an environmental input is found on ENV SOM, the neurons on ENV-BIO SOM linked with the BMU will be tracked at the first stage. At the second stage, the neurons on BIO SOM associated with each of all tracked neurons on ENV-BIO SOM will be highlighted as all possible biological outputs ($PBIO$). After all different environmental inputs (ENV) are applied to the processes, the common neuron(s) (the black colored intersectional neurons on BIO SOM) will be predicted as the target biological output ($TBIO$), which can be described as:

$$TBIO = \cap_{i=0}^{n} PBIO\{ENV_i\}, \tag{1}$$

where n is the number of environmental inputs.

4.2 Weight Vector Linking Method for the Prediction Processes

In order to find the BMUs on each SOM for a given input data, weight vectors of neurons are used to compare their similarity against the input data. From the SOMT structure, combined weight vector, $CW_m = [cw_{m1}\ cw_{m2}\ ...\ cw_{m(e+b)}] \in R^{e+b}(m = 1, 2,...,l$: where l is the number of neurons) for C_n is separated into two sub-weight vectors: $ECW_m = [cw_{m1}\ cw_{m2}\ ...\ cw_{me}] \in R^e$ for E_n and $BCW_m = [cw_{m(e+1)}\ cw_{m(e+2)}\ ...\ cw_{m(e+b)}] \in R^b$ for B_n for the corresponding neurons between the internal and the external SOMs.

Figure 2(b) describes the elements used to generate the weight vector linking distance range (LRange), which is applied to each input data to link the most similar neurons with the observed neuron between the SOMs at each prediction stage. For the first stage, two distances (ED_{ic} and ED_{ik}) for EW_i on ENV SOM are calculated respectively with its best matching sub-weight vector (ECW_c)

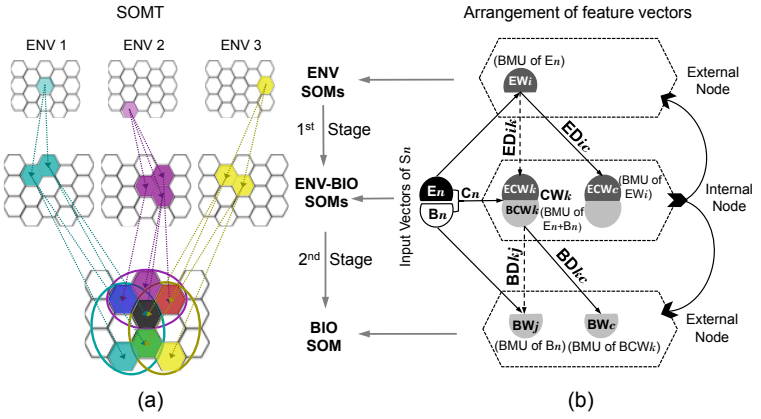


Fig. 2. Structure and algorithm of the SOMT. (a) A visual flow of the SOMT prediction processes. The different colors are used to distinguish each data prediction with tracking arrows. (b) Elements for the weight vector linking method. Unbroken arrows to the BMU and broken arrows to the mapped neuron for the input and the BMU vectors.

and the sub-weight vector of the mapped neuron (ECW_k) on ENV-BIO SOM for each sample data using Euclidean distances such as:

$$ED_{ic} = ||EW_i - ECW_c|| = \sqrt{\sum_{t=1}^{e}(ew_{it} - cw_{ct})^2}. \tag{2}$$

The differences between the two distances for all sample data are analyzed for the first LRange. For the second stage, the distances, BD_{kc} and BD_{kj} are calculated with the same way as the first stage and the differences between them are also analyzed for the second LRange. In this study, the absolute values of the differences for each stage show a normal distribution with the mean value, zero. Using all distributions, a threshold is selected to exclude data when significant increases in the LRange are seen, as determined by the great variations of the diffferences. This results in approximately 1.5 standard deviation of the mean ($\approx$ 86.6%) for the standard difference of all given sample data. The standard difference at each stage will be then added to ED_{ic} and the distance between each tracked neuron on ENV-BIO SOM from the first stage and its BMU on BIO SOM for each LRange. A neuron (weight vector, CW_m) to be linked at the first stage for an input data (weight vector, EW_i) can be described in the following manner:

$$||EW_i - ECW_m|| \leq ED_{ic} + 1.5std\{\cup_{n=1}^{s} S_n\{|ED_{ic} - ED_{ik}|\}\}. \tag{3}$$

The LRange is different for every input data as their BMUs on the SOMs are different. This coupling function places the SOMs in the tracking mode and the neurons, which their weight vectors are linked within the LRange, are tracked.

5 Experimental Results and Discussion

We evaluated the performance of the proposed SOMT for the interactive ecological data predictions with the data relationships. Ecological data for this experiment were acquired from a technical report data series of the U.S. Geographical Survey's National Water-Quality Assessment Program [5]. A total of 146 sample data were chosen with 4 ecological domain data sets. Each data set was formed with 5 components[2] by considering the most influential factors and indicators for ecological data analyses [3,13,14]. Among 146 sample data, 130 were used to train the SOMT, whereas the remaining were used to test the trained model. All data sets were proportionally normalized between 0 and 1.

Four external SOMs were trained for 3 environmental data sets of physical (PHY), chemical (CHE) and land use (LAN) domains and for a biological (BIO) data set. Three internal SOMs were also trained for combined PHY-BIO, CHE-BIO and LAN-BIO data. Each map size (the number of neurons) was selected by considering the minimum value of quantization and topological errors [8,17]. The selected sizes were 10×12 (120 neurons) for all external SOMs and 12×14 (168 neurons) for all internal SOMs. The initial learning rate of 0.05 and 1000 learning iterations were applied to all seven maps. Similar patterns of neurons on each external SOM were clustered by U-matrix and K-means methods with the lowest Davies-Bouldin Index (DBI) [13]. The clusters or component planes of each SOM can be used for the purpose of explaining data relationships in the prediction processes. The internal SOMs were not clustered since they were used to link the external SOMs. The standard difference for the LRange between each external and internal SOMs was analyzed with the value of around 0.1.

A trained sample data, labelled with "D24", was selected to demonstrate the prediction processes of the SOMT (Figure 3). Initially, each BMU for each environmental input of the sample data on PHY, CHE and LAN SOMs was highlighted. At the first stage, the linked neurons on each internal SOM were tracked from the BMU on each ENV SOM. At the second stage, the linked neurons on BIO SOM were predicted from each of the tracked neurons on each internal SOM. From the prediction processes, significantly different BIO outputs in different clusters from the observed BIO output (neuron with label, "D24" in cluster VI on BIO SOM) were predicted by PHY and LAN inputs. The final 4 target neurons on BIO SOM were intersected by all three ENV inputs, and they were highlighted in the same cluster with the observed neuron showing the most similar biological profile.

In this experiment, the SOMT generated the 'one-to-many' and the 'many-to-one' prediction hypotheses for ecological data and allowed the effective visual inspection of the relationships through the processes. Comparing such different

[2] Shredders(%), Filtering-Collectors(%), Collector-Gathers(%), Scrapers(%) and Predators(%) for biological data set; Elevation(m), Slope(%), Stream Order, Embeddedness(%) and Water Temperature($^{\circ}C$) for physical data set; Dissolved Oxygen(mg/l), PH, Nitrates(NO3, mg/l), Organic Carbon(mg/l) and Sulfate(SO4, mg/l) for chemical data set; Forest(%), Herbaceous Up Land(%), Wetlands(%), Crop & Pasture Land(%) and Developed Land(%) for land use data set.

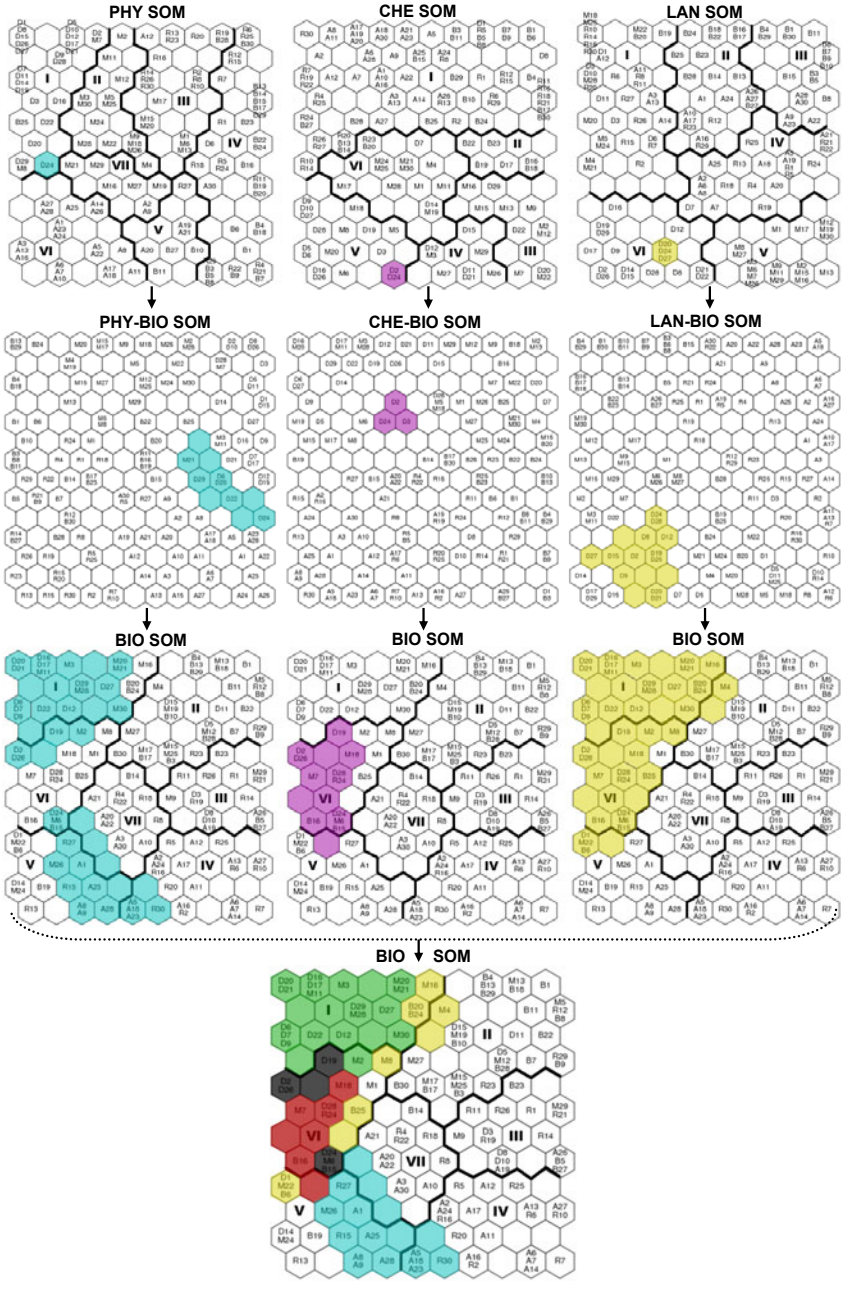


Fig. 3. A visual demonstration of the predictions using the SOMT for a sample data, *"D24"*. Each label in the neurons (BMUs) represents each sampled data. Latin numbers (*I* - *VII*) are used for numbering the clusters on BIO and ENV SOMs. The different colors are used to distinguish each prediction process for different inputs.

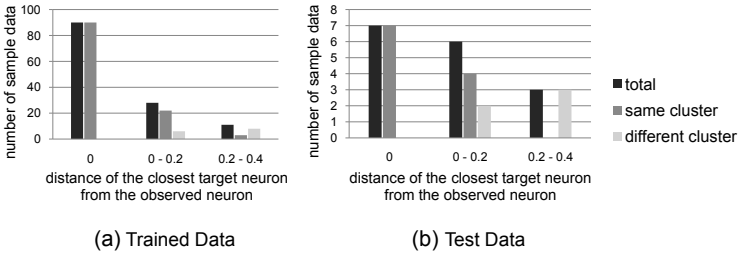


Fig. 4. Histograms of the distance ranges of the closest predicted target neurons from the observed neurons and the number of sample data within the ranges. The closest neurons at the right next to the observed neuron had the distance between *0* and *0.2*.

outputs for each input and the final target output by all inputs may be helpful for the causality analysis of estimating environmental impacts on biological entities.

The predictability of the SOMT was also measured by examining the distances of the predicted target neurons to the observed neuron with their weight vectors. As shown in Figure 4, 89% of the trained data (a) and 69% of the test data (b) predicted most of the final target neurons in the same cluster showing the most similar pattern with the observed neuron from the process results. The SOMT delivered a good result in estimating the common profile of the target outputs although the output values could not easily be quantified with a number of final target neurons. Beyond this experiment, we have begun to carry out more experiemnts with different field data accompanying the sensitivity evaluation of the SOMT for the improved model generalization.

6 Conclusion

In this paper, we proposed an interactive method for nonlinear and multivariate data causality prediction in company with data relationships. The issues of the isolated prediction analysis from the relationship anlaysis using ANNs and the typical 'one-to-one' prediction case of BPN were described. To address the issues, the SOM Tree (SOMT) was constructed with the node SOMs, which were associated by a novel weight vector linking method, for the interactive and transparent prediction processes among different data types. Data relationships were visually inspected through the SOMs and various predicitons were supported by the SOMT processes. Significantly different outputs for an input ('one-to-many' prediction) and the target output by all given inputs ('many-to-one' prediction) were predicted through the processes. The experimental results also showed that the model is highly acceptable for the prediction analysis. This new approach of the SOMT could take into account the variability of nonlinear and multivariate data causality prediction with explaining the complex relationships in the process.

References

1. Aguilera, P.A., Frenich, A.G., Torres, J.A., Castro, H., Vidal, J.L.M., Canton, M.: Application of the kohonen neural network in coastal water management: methodological development for the assessment and prediction of water quality. Water Research 35, 4053–4062 (2001)
2. Chon, T.S., Park, Y.S., Moon, K.H., Cha, E.Y.: Patternizing communities by using an artificial neural network. Ecological Modelling 90, 69–78 (1996)
3. Compin, A., Cereghino, R.: Spatial patterns of macroinvertebrate functional feeding groups in streams in relation to physical variables and land-cover in southwestern france. Landscape Ecology 22, 1215–1225 (2007)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, Inc., New York (2001)
5. Giddings, E.M.P., Bell, A.H., Beaulieu, K.M., Cuffney, T.F., Coles, J.F., Brown, L.R., Fitzpatrick, F.A., Falcone, J., Sprague, L.A., Bryant, W.L., Peppler, M.C., Stephens, C., McMahon, G.: Selected physical, chemical, and biological data used to study urbanizing streams in nine metropolitan areas of the united states, 1999-2004. Technical Report Data Series 423, National Water-Quality Assessment Program, U.S. Geological Survey (2009)
6. Giraudel, J.L., Lek, S.: A comparison of self-organizing map algorithm and some conventional statistical methods for ecological community ordination. Ecological Modelling 146, 329–339 (2001)
7. Kalteh, A.M., Hjorth, P., Berndtsson, R.: Review of the self-organizing map (som) approach in water resources: Analysis, modelling and application. Environmental Modelling and Software 23, 835–845 (2008)
8. Kohonen, T.: Self-Organizing Maps, 3rd edn. Information Sciences. Springer, Heidelberg (2001)
9. Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J.: Som-pak: The self-organizing map program package. Technical Report Version 3.1, SOM Programming Team, Helsinki University of Technology, Helsinki (1995)
10. Lek, S., Guegan, J.F.: Artificial neural networks as a tool in ecological modelling, an introduction. Ecological Modelling 120, 65–73 (1999)
11. Madzarov, G., Gjorgjevikj, D., Chorbev, I.: A multi-class svm classifier utilizing binary decision tree. In: Informatica, pp. 233–241 (2009)
12. Mele, P.M., Crowley, D.E.: Application of self-organizing maps for assessing soil biological quality. Agriculture, Ecosystems and Environment 126, 139–152 (2008)
13. Novotny, V., Virani, H., Manolakos, E.: Self organizing feature maps combined with ecological ordination techniques for effective watershed management. Technical Report 4, Center for Urban Environmental Studies, Northeastern University, Boston (2005)
14. Park, Y.S., Cereghino, R., Compin, A., Lek, S.: Applications of artificial neural networks for patterning and predicting aquatic insect species richness in running waters. Ecological Modelling 160, 265–280 (2003)
15. Sauvage, V.: The t-som (tree-som). In: Sattar, A. (ed.) Canadian AI 1997. LNCS, vol. 1342, pp. 389–397. Springer, Heidelberg (1997)
16. Tran, T.L., Knight, C.G., O'Neill, R.V., Smith, E.R., O'Connell, M.: Self-organizing maps for integrated environmental assessment of the mid-atlantic region. Environmental Management 31, 822–835 (2003)
17. Uriarte, E.A., Martin, F.D.: Topology preservation in som. International Journal of Mathematical and Computer Sciences 1(1), 19–22 (2005)

Document Classification on Relevance: A Study on Eye Gaze Patterns for Reading

Daniel Fahey, Tom Gedeon, and Dingyun Zhu

Research School of Computer Science,
College of Engineering and Computer Science,
The Australian National University,
Acton, Canberra, ACT 0200, Australia
{daniel.fahey,tom.gedeon,dingyun.zhu}@anu.edu.au

Abstract. This paper presents a study that investigates the connection between the way that people read and the way that they understand content. The experiment consisted of having participants read some information on selected documents while an eye-tracking system recorded their eye movements. They were then asked to answer some questions and complete some tasks, on the information they had read. With the intention of investigating effective analysis approaches, both statistical methods and Artificial Neural Networks (ANN) were applied to analyse the collected gaze data in terms of several defined measures regarding the relevance of the text. The results from the statistical analysis do not show any significant correlations between those measures and the relevance of the text. However, good classification results were obtained by using an Artificial Neural Network. This suggests that using advanced learning approaches may provide more insightful differentiations than simple statistical methods particularly in analysing eye gaze reading patterns.

Keywords: Document Classification, Relevance, Gaze Pattern, Reading Behavior, Statistical Analysis, Artificial Neural Networks.

1 Introduction

When people read they display some personal behaviours (usually without noticing it) that break the standard reading paradigm. These differences may be a defining factor on how well a person understands the material that they are reading, or how well they understand information in general. Is it possible to identify a pattern or a key factor, in a person's reading pattern, that can explain how well they will understand the information they are reading? If it is, then a method could be created to measure a person's understanding of some material based entirely on the way that they read that material.

With the motivation of studying eye gaze patterns particularly for reading, an experiment has been conducted to test how well a person can understand the premise for a paper when they are given paragraphs from that paper in a random order. Of the paragraphs that are given only half contain much useful information

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 143–150, 2011.
© Springer-Verlag Berlin Heidelberg 2011

while the other half contain much less. The experimental participants read the paragraphs with their eye gaze being tracked using a computerised eye-tracking system. Questions were asked and some other tasks referring the paragraphs were completed to score a participant's understanding of the original paper. The results of this experiment are expected to be used to try and find if there is some characteristic of a persons gaze pattern that can be attributed to having a better or worse understanding of the information. This could be used to devise a method of testing people for how well they understand information.

2 Eye Gaze for Reading

Apart from the research work on using eye gaze as an input for conventional user interfaces [2], studying human's reading behaviour in terms of their eye gaze is another field with much research effort.

Several algorithms exist to detect whether a user is reading or not based on their eye gaze. One such system is the "Pooled Evidence" system [1] which classifies a user's behaviour into either a scanning mode or a reading mode. An evidence threshold is used to determine how much evidence is required (in points) and different types of reading behaviours are given point values for how much evidence they contribute.

In [4], a thorough review of eye movements in reading and information processing has been conducted with a summary of three interesting examples of eye movement characteristics during reading, which have become important references regarding gaze parameters in reading:

1. When reading English, eye fixations last about 200-250 ms and the mean saccade size is 7-9 letter spaces.
2. Eye movements are influenced by textual and typographical variables, e.g., as text becomes conceptually more difficult, fixation duration increases and saccade length decreases. Factors such as the quality of print, line length, and letter spacing influence eye movements.
3. Eye movements differ somewhat when reading silently from reading aloud: mean fixation durations are longer when reading aloud or while listening to a voice reading the same text than in silent reading.

More recently, new methods based on advanced learning approaches have been proposed to be useful for studying gaze patterns in reading. In [8], a hybrid fuzzy approach for eye gaze pattern recognition has been introduced. This approach combines fuzzy signatures [3] with Levenberg-Marquardt optimization method for recognizing the different eye gaze patterns when a human is viewing faces or text documents. The experimental results show the effectiveness of using this method for the real world case. A further comparison with Support Vector Machines (SVM) also demonstrates that by defining the classification process

in a similar way to SVM, this hybrid approach is able to provide a comparable performance but with a more interpretable form of the learned structure.

Furthermore, a similar method has been introduced in [6] by which detecting the level of engagement in reading based on a person's gaze pattern becomes possible. Through their experimental results, they demonstrate the feasibility of the applying this approach in real-life systems.

3 The Experiment

In order to analyse different reading patterns an experiment was designed. The experiment involved reading a series of paragraphs and then answering some questions about those paragraphs.

3.1 Experiment Design

In all there were ten paragraphs for the participants to read. Seven of the paragraphs were taken from a selected paper [7]. The remaining three paragraphs were written by students who were required to write about the paper for course work. Five of the paragraphs from the paper were chosen for the amount of useful information that was contained within. The other two paragraphs from the paper and the three student paragraphs were chosen because of their generality and lack of useful information. Care was taken to make sure that this fact was not obvious.

The paragraphs were presented to different participants in different orders to prevent any specific paragraph ordering from affecting the results. The paragraphs all come from different places in the paper or from a completely different source altogether (the student's paragraphs). As well as being presented in different orders, the overall composition of the paragraphs became very convoluted. This was an experiment design choice to help show which participants could look at the bigger picture even when the information is out of place and scattered. The participants were given 90 seconds to read each paragraph.

After reading the ten paragraphs, the participants were asked to answer five multiple choice questions on the material. These questions asked about the content of the five paragraphs that contained the most relevant information. Furthermore, they were asked to write describe the paper in one sentence. Only one sentence was asked for, to not inundate the participant with a writing task. Then they were asked to rank the paragraphs from the one with the most useful information for completing the questions, as number one, and the one with the least information, as number ten. All the data were used to analyse how well they had understood the material that was presented to them. Then the utility of their reading patterns and characteristics could be assessed.

3.2 Experimental Setup

During the experiments the participants read all the paragraphs off a screen which was connected to the same computer that was recording their eye movements.

The computer was a standard desktop machine that was running Windows XP. The eye tracking system that was connected to the computer was provided by Seeingmachines with FaceLab V4.5 software [5].

As shown in Fig. 1, the computer had two screens connected to it, one for controlling and monitoring the experiment and a 19 inch screen with a resolution of 1280 by 1024 for the participants to read the paragraphs and questions off. Before the experiment could begin, the system was calibrated for each participant. All the paragraphs and questions were set to the same resolution so no scaling was required. The entire system was housed on a cart that had a mounted chin rest to help the participants keep their head still. Although the chin rest helped to keep the participants head still there were still times when the gaze tracking system would lose its target, usually if the participant started to squint when reading the bottom of the screen (when tracking was lost no data points were recorded and so it can be identified where this happened and is taken into account in the analysis).



Fig. 1. The Setup for the Reading Experiment

3.3 Participants

Altogether 18 volunteers from a local university participated in the experiment, of them 3 were removed because of the poor results, i.e. the gaze tracker recorded only noise or nothing.

4 Analysis and Results

4.1 Gaze Points to Fixations

A person's gaze is characterised by two behaviours, fixations and saccades [2]. Fixations being the time when a person focuses on an object and they move that object into view of their fovea (the part of the eye with the most photosensitive cells). A saccade is the high-speed, ballistic movement of the eye when it is between fixations. It is reasonable to display everything in fixations (and saccades, although saccades are not really displayed because they are just the movement between the really meaningful data).

To break the gaze points into fixations an approximate method was used. As shown in Fig.2, the fixations are represented as circles that are centred at the average position of all the gaze points that are contained within them and their radius is determined by the length of time that the participant spent in that fixation. Thin lines are drawn between the fixations and could be considered saccades although, they are only there to show an observer which fixation comes next and they do not take into account any of the gaze points in the saccades. The gaze points in the saccades are essentially omitted. The same colouring scheme applies on the fixations as did on the gaze points, the colour gets lighter as time passes.

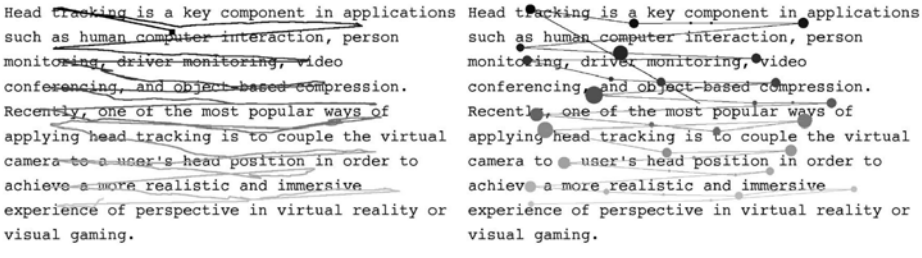


Fig. 2. Gaze Points / Lines (left) vs Fixations (right) Generated from the Collected Gaze Data

4.2 Scoring the Participants

The evaluation of the participants was a step that was inherent in the experiment. It was the purpose of asking the questions, and having the participants write a sentence and rank the paragraphs. The experiment was designed so that the participants could be scored using the following guidelines:

Paragraph Ranking: ten paragraphs and one point would be awarded for each paragraph that was correctly ranked in the correct half.
Multiple Choice: one point would be awarded to each correct answer.

Sentence Writing: a possible three points awarded for the sentence regarding whether participants have mentioned the key content of the paper.

The participant could have received a score of up to 18 points. These scores would allow the participants understanding to be quantified so that the ones that understood better could be identified. In the end, the highest scoring participant received a score of 16, the lowest scoring participant received a score of 4, the mean score was 9.6 with a standard deviation of 3.29.

4.3 Statistical Analysis

Before the statistical analysis, a few measurements were taken about the way that the participants read. These measurements were taken as averages across entire slides. The measurements that were taken were:

1. Time taken to read a slide.
2. Horizontal distance between fixations.
3. Vertical distance between fixations.
4. Number of gaze points per slide.
5. Number of fixations per slide.
6. Length of fixations.

These measurements were plotted against scores to try and find trends. There were some slight trends although none of them were statistically significant. It seems that simple statistical analysis did not show any real correlation between the simple measurements and the scores.

4.4 Further Analysis by ANN

To look into the merits of using more advanced analysis techniques on the data, a neural network was trained to determine whether a given paragraph was relevant or irrelevant. To do this only the data from the gaze patterns of the paragraphs was used. The neural network was back propagation trained and its inputs consisted of the measurements that were taken above except on the individual paragraphs. The neural network had six hidden nodes and one output, which was the class for that given paragraph, that the inputs corresponded to, was relevant or irrelevant. The neural network was trained with 60% of the data while 20% was used to generalise the network and prevent over-fitting and the last 20% was used as the test data.

The neural network produced good results (see Fig.3) with a correct classification rate of approximately 86% (assuming that there is no undecided class, so all points that are on the correct side of 0.5 are considered correct). Training this neural network was only an example of how learning algorithms can be used to analyse this data.

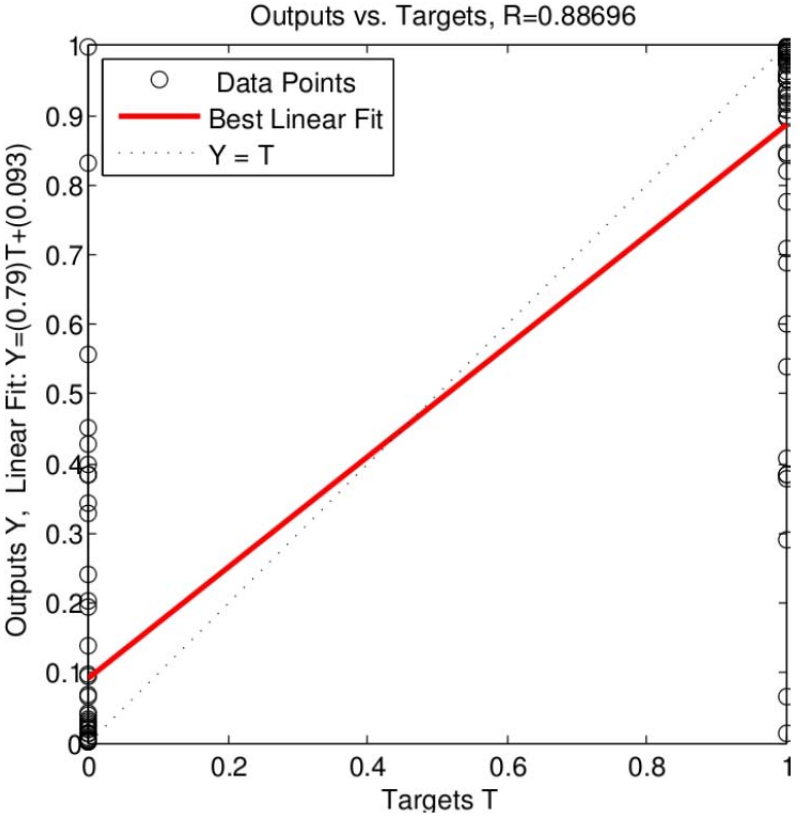


Fig. 3. A graph of the results for the neural network. The dots down the sides correspond to the paragraphs. The ones on the left are the irrelevant paragraphs and the ones on the right are the relevant paragraphs. Their location shows their class as one or the other according to this model. So, the irrelevant paragraphs should be near the bottom and the relevant ones should be near the top. The solid line that runs across the graph is the line of best fit between all the dots. The dotted line that runs across the graph is the ideal solution (where every paragraph is correctly classed).

5 Discussion

From the results, it shows that using classical statistical methods, we could hardly find any significant correlations between the measures we defined in terms of the gaze data and the scores of the participants in the reading experiment. However, good classification results were generated for discriminating between relevant and irrelevant paragraphs by training a simple artificial neural network with the same input data from the defined measures. This implies the potential advantages of using advanced learning approaches especially for analysing eye gaze patterns in reading. These approaches might be more useful in studying more detailed information within the gaze data than applying traditional methods, which also requires further investigations and comparisons.

Future studies could include using the same method to see if the learning algorithms could determine which questions a participant will get right or wrong, or perhaps even predict which ordering a participant will order their paragraphs in. But what would be much more useful for trying to quantify a participants understanding would be to train a learning algorithm on the values of the gaze points, or fixations themselves.

References

1. Compbell, C.S., Maglio, P.P.: A Rbust Algorithm for Reading Detection. In: 2001 Workshop on Perceptive User Interfaces, vol. 15, pp. 1–7. ACM (2001)
2. Jacob, R.J.K.: The Use of Eye Movements in Human-computer Interaction Techniques: What You Look at is What You Get. ACM Transactions on Information Systems 9(2), 152–169 (1991)
3. Koczy, L.T., Vamos, T., Biro, G.: Fuzzy Signatures. In: Proceedings of the 4th Meeting of the Euro Working Group on Fuzzy Sets and the 2nd International Conference on Soft and Intelligent Computing (EUROPUSE-SIC 1999), Budapest, Hungary, pp. 210–217 (1999)
4. Rayner, K.: Eye Movements in Reading and Information Processing: 20 Years of Research. Psychological Bulletin 124(3), 372–422 (1998)
5. Seeingmachines, Inc: FaceLAB (2011), http://www.seeingmachines.com/faceLAB.html
6. Vo, T., Mendis, B.S.U., Gedeon, T.: Gaze Pattern and Reading Comprehension. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) ICONIP 2010 Part II. LNCS, vol. 6444, pp. 124–131. Springer, Heidelberg (2010)
7. Zhu, D., Gedeon, T., Taylor, K.: Keyboard before Head Tracking Depresses User Success in Remote Camera Control. In: Gross, T., Gulliksen, J., Kotzé, P., Oestreicher, L., Palanque, P., Prates, R.O., Winckler, M. (eds.) INTERACT 2009. LNCS, vol. 5727, pp. 319–331. Springer, Heidelberg (2009)
8. Zhu, D., Mendis, B.S.U., Gedeon, T., Asthana, A., Goecke, R.: A Hybrid Fuzzy Approach for Human Eye Gaze Pattern Recognition. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008. LNCS, vol. 5507, pp. 655–662. Springer, Heidelberg (2009)

Multi-Task Low-Rank Metric Learning Based on Common Subspace

Peipei Yang, Kaizhu Huang, and Cheng-Lin Liu

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences,
Beijing, China 100190
{ppyang,kzhuang,liucl}@nlpr.ia.ac.cn

Abstract. Multi-task learning, referring to the joint training of multiple problems, can usually lead to better performance by exploiting the shared information across all the problems. On the other hand, metric learning, an important research topic, is however often studied in the traditional single task setting. Targeting this problem, in this paper, we propose a novel multi-task metric learning framework. Based on the assumption that the discriminative information across all the tasks can be retained in a low-dimensional common subspace, our proposed framework can be readily used to extend many current metric learning approaches for the multi-task scenario. In particular, we apply our framework on a popular metric learning method called Large Margin Component Analysis (LMCA) and yield a new model called multi-task LMCA (mtLMCA). In addition to learning an appropriate metric, this model optimizes directly on the transformation matrix and demonstrates surprisingly good performance compared to many competitive approaches. One appealing feature of the proposed mtLMCA is that we can learn a metric of low rank, which proves effective in suppressing noise and hence more resistant to over-fitting. A series of experiments demonstrate the superiority of our proposed framework against four other comparison algorithms on both synthetic and real data.

Keywords: Multi-task Learning, Metric Learning, Low Rank, Subspace.

1 Introduction

Multi-task learning (MTL), referring to the joint training of multiple problems, has recently received considerable attention [2,4,1,8,14]. If the different problems are closely related, MTL can usually lead to better performance by propagating discriminative information among tasks. For a better illustration of MTL, we borrow the well-known example from speech recognition [5]. Apparently, different persons pronounce the same words in a different way, which could be influenced by their gender, accent, nationality or other characteristics. Each individual speaker can then be viewed as different problems or tasks that are closely related to each other. Joint training of these different problems could lead to

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 151–159, 2011.
© Springer-Verlag Berlin Heidelberg 2011

better generalization performance for each individual task. This approach proves very effective especially when few samples can be obtained for certain problems.

On the other hand, distance or metric learning has been widely studied in machine learning due to its importance in many machine learning tasks [13,6,12,7,11,3]. However, most of the current metric learning methods are single-task oriented. They are incapable of taking advantages of multi-task learning. When the number of training samples in some tasks is small, they usually fail to learn a good metric and hence cannot deliver better classification or clustering performance.

In this paper, aiming to solve this problem, we propose a general multi-task metric learning framework. Based on the assumption that the discriminative information across all the tasks can be retained in a low-dimensional common subspace, our proposed framework can be readily used to extend many current metric learning approaches for multi-task learning. In particular, we apply our framework on a popular metric learning method called Large Margin Component Analysis (LMCA) [11] and yield a new model called multi-task LMCA (mtLMCA). In addition to learning an appropriate metric, this model optimizes directly on the transformation matrix and demonstrates surprisingly good performance compared to many competitive approaches. One appealing feature of the proposed mtLMCA is that we can learn a metric of low rank, which can suppress noise effectively and hence be more resistant to over-fitting.

We note that Parameswaran et al. recently proposed a multi-task metric learning method called mtLMNN based on the Large Margin Metric Learning (LMNN) model [9]. Following [4], mtLMNN assumes that the distance metric for each task is combined by a common metric with a task-specific metric. This approach suffers from two shortcomings. (1) It cannot directly learn a low-rank metric, which however proves critical for resisting overfitting. (2) It is computationally more complicated, especially when the dimensionality is high. Denote the task number and the data dimensionality are t and D respectively. There are $(t+1)D^2$ parameters to be optimized in mtLMNN. In comparison, there are merely $Dd+td^2$ parameters in our approach. Here $d \ll D$ represents the dimensionality of the common subspace. Finally, later experimental results show that our proposed approach consistently outperforms mtLMNN in many datasets.

The rest of this paper is organized as follows. In Section 2, we introduce our novel framework in details. In Section 3, we evaluate our framework on four datasets. Finally, we set out the conclusion in Section 4.

2 Multi-Task Low-Rank Metric Learning

In this section, we first present the notation and the problem definition.We then introduce our proposed multi-task metric learning framework in details.

2.1 Notation and Problem Definition

Assume that there are T related tasks. For the t-th task, we are given a training data set $\mathcal{S}_t$ containing N_t D-dimensional data points $\mathbf{x}_{tk} \in \mathbb{R}^D, k = 1, 2, \ldots, N_t$.

The basic target of multi-task metric learning is to learn an appropriate distance metric f_t for each task t utilizing all the information from the joint training set $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_T\}$. The distance metric f_t should satisfy extra constraints on a set of triplets $\mathcal{T}_t = \{(i,j,k)|f_t(\mathbf{x}_{ti}, \mathbf{x}_{tj}) \leq f(\mathbf{x}_{ti}, \mathbf{x}_{tk})\}$ [10].[1] These constraints can force similar data pairs, e.g.,$\mathbf{x}_{ti}$ and $\mathbf{x}_{tj}$ to stay closer than dissimilar pairs e.g.,$\mathbf{x}_{ti}$ and $\mathbf{x}_{tk}$ with the new distance metric f_t. We denote the set of all the similar and dissimilar pairs appearing in $\mathcal{T}_t$ as $\mathcal{S}_t$ and $\mathcal{D}_t$ respectively.

In the context of low-rank metric learning, f_t is assumed to be a linear transformation $L_t : \mathbb{R}^D \rightarrow \mathbb{R}^d$ (with $d \ll D$ for obtaining a low rank) such that $\forall(i,j,k) \in \mathcal{T}_t, ||\hat{\mathbf{x}}_{ti} - \hat{\mathbf{x}}_{tj}||_2^2 \leq ||\hat{\mathbf{x}}_{ti} - \hat{\mathbf{x}}_{tk}||_2^2$, with $\hat{\mathbf{x}}_{tk} = L_t\mathbf{x}_{tk}$, i.e., the distance function can be defined as $f_t(\mathbf{x}_{ti}, \mathbf{x}_{tj}) = \text{dist}_{L_t}(\mathbf{x}_{ti}, \mathbf{x}_{tj}) \triangleq \mathbf{x}_{t,ij}^\top L_t^\top L_t \mathbf{x}_{t,ij}$ where $\mathbf{x}_{t,ij} = \mathbf{x}_{ti} - \mathbf{x}_{tj}$. For brevity, we also write $f_t(\mathbf{x}_{ti}, \mathbf{x}_{tj}) = f_{t,ij}(L_t)$.

The loss involved in task t (defined as l_t) is hence determined by the distance function f_t (or transformation L_t) and the pairs appearing in triplet set $\mathcal{T}_t$:

$$l_t = \epsilon_t(L_t) = \epsilon_t(\{f_{t,ij}(L_t)\}),\ (i,j) \in \mathcal{S}_t \cup \mathcal{D}_t,$$

where ϵ_t is any available loss function. Hence the overall loss involved in all the tasks can be written as

$$l(\{L_t\}) = \sum_t l_t = \sum_t \epsilon_t(L_t). \tag{1}$$

In order to utilize the correlation information among tasks, we assume that the discriminative information embedded in L_t can be retained in a common subspace L_0. We will introduce the detailed framework in the next subsection.

2.2 Multi-Task Framework for Low-Rank Metric Learning

Let the "economy size" singular value decomposition (SVD) of the $d \times D$ transformation matrix be $L_t = U_t S_t V_t^\top$, where S_t is an $r \times r$ diagonal matrix with the non-zero singular values. Then we have

$$\begin{aligned} \text{dist}_{L_t}(\mathbf{x}_{ti}, \mathbf{x}_{tj}) =& \mathbf{x}_{t,ij}^\top V_t S_t^\top U_t^\top U_t S_t V_t^\top \mathbf{x}_{t,ij} = \left(V_t^\top \mathbf{x}_{t,ij}\right)^\top \left(S_t^\top S_t\right) \left(V_t^\top \mathbf{x}_{t,ij}\right) \\ =& \hat{\mathbf{x}}_{t,ij}^\top (S_t^\top S_t) \hat{\mathbf{x}}_{t,ij} = \text{dist}_{S_t}(\hat{\mathbf{x}}_{ti}, \hat{\mathbf{x}}_{tj}). \end{aligned} \tag{2}$$

Equation (2) means that the distance of any two points $\mathbf{x}_{ti}, \mathbf{x}_{tj}$ defined by L_t in the original space is equivalent to the distance of their projections $\hat{\mathbf{x}}_{ti}, \hat{\mathbf{x}}_{tj}$ defined by S_t in the low-rank subspace $\mathbf{R}(V_t) = \mathbf{R}(L_t^\top)$.

Based on the discussion above, we can model the task relationship with the major assumption: there exists an L_0 defining the common subspace to make that $\mathbf{R}(L_t^\top) \subseteq \mathbf{R}(L_0^\top),\ t = 1, \ldots, T$. This means that the distance information for all the tasks can be retained in a low-dimensional common subspace $\mathbf{R}(L_0^\top)$. Therefore, we can use a $d \times D$ matrix L_0 to represent the common subspace for all the tasks, and try to exploit a $d \times d$ square matrix R_t to learn a specific metric in the subspace for each task. This leads the learned metric for task t can be written as $L_t = R_t L_0$.

[1] Other settings could be also used.

With the constraint above, we then would like to minimize the overall loss l defined in Eq. (1). The final optimization problem of multi-task low-rank metric learning can be written as follows:

$$\min_{L_0,\{R_t\}} l(L_0,\{R_t\}) = \sum_t \epsilon_t(R_t L_0) = \sum_t \epsilon_t(\{f_{t,ij}(R_t L_0)\}),\ (i,j) \in \mathcal{S}_t \cup \mathcal{D}_t, \quad (3)$$

where $f_{t,ij}(R_t L_0) = \mathbf{x}_{t,ij}^\top L_0^\top R_t^\top R_t L_0 \mathbf{x}_{t,ij}$.

2.3 Optimization

In the following, we try to adopt the gradient descent method to solve the optimization problem (3).

$$\begin{aligned}\frac{\partial \epsilon_t}{\partial L_t} &= \sum_i \left(\frac{\partial \epsilon_t}{\partial f_{t,ij}} \cdot \frac{\partial f_{t,ij}}{\partial L_t}\right) = \sum_{i,j} \left(\frac{\partial \epsilon_t}{\partial f_{t,ij}} \cdot 2L_t \mathbf{x}_{t,ij}\mathbf{x}_{t,ij}^\top\right) \\ &= 2L_t \sum_{i,j} \left(\frac{\partial \epsilon_t}{\partial f_{t,ij}} \cdot \mathbf{x}_{t,ij}\mathbf{x}_{t,ij}^\top\right). \end{aligned} \quad (4)$$

Since $\frac{\partial f_{t,ij}}{\partial L_0} = 2R_t^\top R_t L_0 \mathbf{x}_{t,ij}\mathbf{x}_{t,ij}^\top$, the gradient can then be calculated as

$$\begin{aligned}\frac{\partial l}{\partial L_0} &= \sum_t \frac{\partial \epsilon_t}{\partial L_0} = \sum_t \left(2R_t^\top R_t L_0 \sum_i \left(\frac{\partial \epsilon_t}{\partial f_{t,ij}} \cdot \mathbf{x}_{t,ij}\mathbf{x}_{t,ij}^\top\right)\right) = \sum_t \left(2R_t^\top R_t L_0 \Delta_t\right) \\ \frac{\partial l}{\partial R_t} &= \frac{\partial \epsilon_t}{\partial R_t} = 2R_t \sum_{i,j} \left(\frac{\partial \epsilon_t}{\partial f_{t,ij}} \cdot (L_0\mathbf{x}_{t,ij})(L_0\mathbf{x}_{t,ij})^\top\right) = 2R_t L_0 \Delta_t L_0^\top, \end{aligned} \quad (5)$$

where

$$\Delta_t = \sum_{i,j} \left(\frac{\partial \epsilon_t}{\partial f_{t,ij}} \cdot \mathbf{x}_{t,ij}\mathbf{x}_{t,ij}^\top\right). \quad (6)$$

With (4)-(6), we can easily use the gradient descend method to optimize the L_0 and R_t and hence obtain the final low-rank metric for each task.

2.4 Special Case

In this section, we show how to apply our multi-task low-rank metric learning framework to a specific metric learning method. We take the LMCA [11] as a typical example and develop a Multi-task LMCA model.[2]

In LMCA, for each sample, some nearest neighbors with the same label are defined as *target neighbors*, which are assumed to have established a perimeter such that differently labeled samples should not invade. Those differently labeled samples invading this perimeter are referred to as *impostors* and the goal of learning is to minimize the number of impostors. The difference between

[2] Note that it is straightforward to extend our framework to the other metric learning models which optimize the objective function with the transformation matrix.

LMCA and LMNN is that LMCA optimizes the transformation matrix L_t while LMNN optimizes the Mahalanobis matrix $M_t = L_t^\top L_t$. Given n input examples $\mathbf{x}_{t1}, \ldots, \mathbf{x}_{tn}$ in $\mathbb{R}^D$ and their corresponding class labels $y_{t1}, \ldots, y_{tn}$, the loss function with respect to transformation matrix L_t is

$$\begin{aligned}\epsilon_t(L_t) =&(1-\mu)\sum_{i,j\rightsquigarrow i}\|L_t(\mathbf{x}_{ti}-\mathbf{x}_{tj})\|^2+\\ &\mu\sum_{i,j\rightsquigarrow i,k}(1-y_{t,ik})\mathbf{h}\left(\|L(\mathbf{x}_{ti}-\mathbf{x}_{tj})\|^2-\|L(\mathbf{x}_{ti}-\mathbf{x}_{tk})\|^2+1\right),\end{aligned} \tag{7}$$

where $y_{t,ik} \in \{0,1\}$ is 1 iff $y_{ti} = y_{tk}$, and $\mathbf{h}(s) = \max(s, 0)$ is the hinge function. Minimizing $\epsilon_t(L_t)$ can be implemented using the gradient-based method. Define $\mathcal{T}_t$ as the set of triples which trigger the hinge loss:

$$(i,j,k) \in \mathcal{T}_t \ \textit{iff}\ \|L_t(\mathbf{x}_{ti}-\mathbf{x}_{tj})\|^2-\|L_t(\mathbf{x}_{ti}-\mathbf{x}_{tk})\|^2+1>0.$$

Substituting the transformation matrix of task-t with $L_t = R_t L_0$ and the loss ϵ_t in (6) with (7), we have

$$\begin{aligned}\Delta_t =&(1-\mu)\sum_{i,j\rightsquigarrow i}(\mathbf{x}_{ti}-\mathbf{x}_{tj})(\mathbf{x}_{ti}-\mathbf{x}_{tj})^\top+\\ &\mu\sum_{(i,j,k)\in\mathcal{T}_t}(1-y_{t,ik})\left[(\mathbf{x}_{ti}-\mathbf{x}_{tj})(\mathbf{x}_{ti}-\mathbf{x}_{tj})^\top-(\mathbf{x}_{ti}-\mathbf{x}_{tk})(\mathbf{x}_{ti}-\mathbf{x}_{tk})^\top\right].\end{aligned}$$

Using Δ_t, the gradient can be calculated with Eq. (5).

3 Experiments

In this section, we first illustrate our proposed multi-task method on a synthetic data set. We then conduct extensive evaluations on three real data sets in comparison with four competitive methods.

3.1 Illustration on Synthetic Data

In this section, we take the example of concentric circles in [6] to illustrate the effect of our multi-task framework. Assume there are T classification tasks where the samples are distributed in the 3-dimensional space and there are c_t classes in the t-th task. For all the tasks, there exists a **common** 2-dimensional subspace (plane) in which the samples of each class are distributed in an elliptical ring centered at zero. The third dimension orthogonal to this plane is merely Gaussian noise. The samples of randomly generated 4 tasks were shown in the first column of Fig. 1. In this example, there are 2, 3, 3, 2 classes in the 4 tasks respectively and each color corresponds to one class. The circle points and the dot points are respectively training samples and test samples with the same distribution. Moreover, as the Gaussian noise will largely degrade the distance calculation

in the original space, we should try to search a low-rank metric defined in a low-dimensional subspace.

We apply our proposed mtLMCA on the synthetic data and try to find a reasonable metric by unitizing the correlation information across all the tasks. We project all the points to the subspace which is defined by the learned metric. We visualize the results in Fig. 1. For comparison, we also show the results obtained by the traditional PCA, the individual LMCA (applied individually on each task). Clearly, we can see that for task 1 and task 4, PCA (column 3) found improper metrics due to the large Gaussian noise. For individual LMCA (column 4), the samples are mixed in task 2 because the training samples are not enough. This leads to an improper metric in task 2. In comparison, our proposed mtLMCA (column 5) perfectly found the best metric for each task by exploiting the shared information across all the tasks.

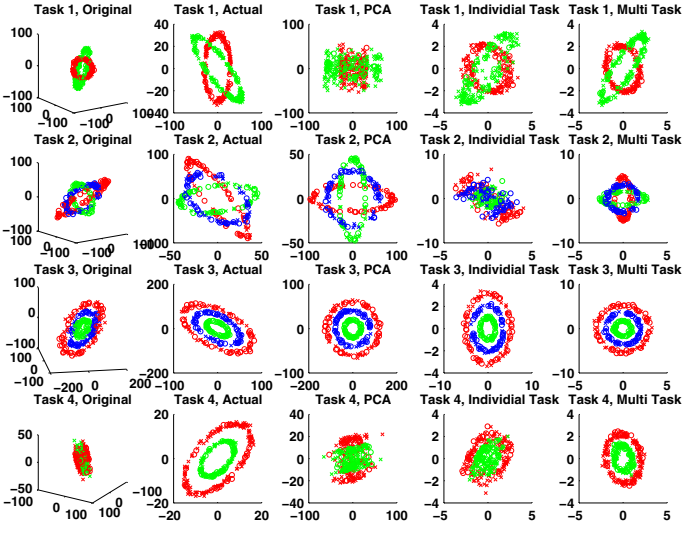


Fig. 1. Illustration for the proposed multi-task low-rank metric learning method (The figure is best viewed in color)

3.2 Experiment on Real Data

We evaluate our proposal mtLMCA method on three multi-task data sets. (1). *Wine Quality data*[3] is about wine quality including 1,599 red samples and 4,898 white wine samples. The labels are given by experts with grades between 0 and 10. (2). *Handwritten Letter Classification data* contain handwritten words. It consists of 8 binary classification problems: c/e, g/y, m/n, a/g, i/j, a/o, f/t, h/n. The features are the bitmap of the images of written letters. (3). USPS data[4] consist of 7,291 16×16 grayscale images of digits $0 \sim 9$ automatically scanned from

[3] http://archive.ics.uci.edu/ml/datasets/Wine+Quality

[4] http://www-i6.informatik.rwth-aachen.de/~keysers/usps.html

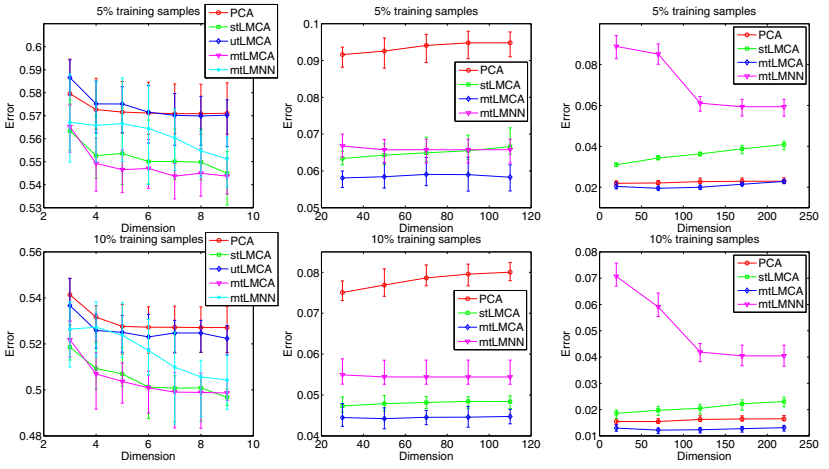


Fig. 2. Test results on 3 datasets (one column respect to one dataset): (1)Wine Quality; (2)Handwritten; (3)USPS. Two rows correspond to 5% and 10% training samples

envelopes by the U.S. Postal Service. The features are then the 256 grayscale values. For each digit, we can get a two-class classification task in which the samples of this digit represent the positive patterns and the others negative patterns. Therefore, there are 10 tasks in total. For the label-compatible dataset, i.e., the Wine Quality data set, we compare our proposed model with PCA, single-task LMCA (stLMCA), uniform-task LMCA (utLMCA)[5], and mtLMNN [9]. For the remaining two label-incompatible tasks, since the output space is different depending on different tasks, the uniform metric can not be learned and the other 3 approaches are then compared with mtLMCA.

Following many previous work, we use the category information to generate relative similarity pairs. For each sample, the nearest 2 neighbors in terms of Euclidean distance are chosen as target neighbors, while the samples sharing different labels and staying closer than any target neighbor are chosen as imposers. For each data set, we apply these algorithms to learn a metric of different ranks with the training samples and then compare the classification error rate on the test samples using the *nearest neighbor* method. Since mtLMNN is unable to learn a low-rank metric directly, we implement an eigenvalue decomposition on the learned Mahalanobis matrix and use the eigenvectors corresponding to the d largest eigenvalues to generate a low-rank transformation matrix. The parameter μ in the objective function is set to 0.5 empirically in our experiment. The optimization is initialized with $L_0 = \mathbf{I}_{d\times D}$ and $R_t = \mathbf{I}_d, t = 1, \ldots, T$, where $\mathbf{I}_{d\times D}$ is a matrix with all the diagonal elements set to 1 and other elements set to 0. The optimization process is terminated if the relative difference of the objective function is less than η, which is set to 10^{-5} in our experiment. We choose

[5] The uniform-task approach gathers the samples in all tasks together and learns a uniform metric for all tasks.

randomly 5% and 10% of samples respectively for each data set as training data while leaving the remaining data as test samples. We run the experiments 5 times and plot the average error, the maximum error, and the minimum error for each data set.

The results are plotted in Fig. 2 for the three data sets. Obviously, in all the dimensionality, our proposed mtLMCA model performs the best across all the data sets whenever we use 5% or 10% training samples. The performance difference is even more distinct in Handwritten Character and USPS data. This clearly demonstrates the superiority of our proposed multi-task framework.

4 Conclusion

In this paper, we proposed a new framework capable of extending metric learning to the multi-task scenario. Based on the assumption that the discriminative information across all the tasks can be retained in a low-dimensional common subspace, our proposed framework can be easily solved via the standard gradient descend method. In particular, we applied our framework on a popular metric learning method called Large Margin Component Analysis (LMCA) and developed a new model called multi-task LMCA (mtLMCA). In addition to learning an appropriate metric, this model optimized directly on a low-rank transformation matrix and demonstrated surprisingly good performance compared to many competitive approaches. We conducted extensive experiments on one synthetic and three real multi-task data sets. Experiments results showed that our proposed mtLMCA model can always outperform the other four comparison algorithms.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (NSFC) under grants No. 61075052 and No. 60825301.

References

1. Argyriou, A., Evgeniou, T.: Convex multi-task feature learning. Machine Learning 73(3), 243–272 (2008)
2. Caruana, R.: Multitask learning. Machine Learning 28(1), 41–75 (1997)
3. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proceedings of the 24th International Conference on Machine Learning, pp. 209–216 (2007)
4. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 109–117 (2004)
5. Fanty, M.A., Cole, R.: Spoken letter recognition. In: Advances in Neural Information Processing Systems, p. 220 (1990)
6. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood component analysis. In: Advances in Neural Information Processing Systems (2004)
7. Huang, K., Ying, Y., Campbell, C.: Gsml: A unified framework for sparse metric learning. In: Ninth IEEE International Conference on Data Mining, pp. 189–198 (2009)

8. Micchelli, C.A., Ponti, M.: Kernels for multi-task learning. In: Advances in Neural Information Processing, pp. 921–928 (2004)
9. Parameswaran, S., Weinberger, K.Q.: Large margin multi-task metric learning. In: Advances in Neural Information Processing Systems (2010)
10. Rosales, R., Fung, G.: Learning sparse metrics via linear programming. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 367–373 (2006)
11. Torresani, L., Lee, K.: Large margin component analysis. In: Advances in Neural Information Processing, pp. 505–512 (2007)
12. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. The Journal of Machine Learning Research 10 (2009)
13. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: Advances in Neural Information Processing Systems, vol. 15, pp. 505–512 (2003)
14. Zhang, Y., Yeung, D.Y., Xu, Q.: Probabilistic multi-task feature selection. In: Advances in Neural Information Processing Systems, pp. 2559–2567 (2010)

Reservoir-Based Evolving Spiking Neural Network for Spatio-temporal Pattern Recognition

Stefan Schliebs[1], Haza Nuzly Abdull Hamed[1,2], and Nikola Kasabov[1,3]

[1] KEDRI, Auckland University of Technology, New Zealand
{sschlieb,hnuzly,nkasabov}@aut.ac.nz
www.kedri.info
[2] Soft Computing Research Group, Universiti Teknologi Malaysia
81310 UTM Johor Bahru, Johor, Malaysia
haza@utm.my
[3] Institute for Neuroinformatics, ETH and University of Zurich, Switzerland

Abstract. Evolving spiking neural networks (eSNN) are computational models that are trained in an one-pass mode from streams of data. They evolve their structure and functionality from incoming data. The paper presents an extension of eSNN called reservoir-based eSNN (reSNN) that allows efficient processing of spatio-temporal data. By classifying the response of a recurrent spiking neural network that is stimulated by a spatio-temporal input signal, the eSNN acts as a readout function for a Liquid State Machine. The classification characteristics of the extended eSNN are illustrated and investigated using the LIBRAS sign language dataset. The paper provides some practical guidelines for configuring the proposed model and shows a competitive classification performance in the obtained experimental results.

Keywords: Spiking Neural Networks, Evolving Systems, Spatio-Temporal Patterns.

1 Introduction

The desire to better understand the remarkable information processing capabilities of the mammalian brain has led to the development of more complex and biologically plausible connectionist models, namely spiking neural networks (SNN). See [3] for a comprehensive standard text on the material. These models use trains of spikes as internal information representation rather than continuous variables. Nowadays, many studies attempt to use SNN for practical applications, some of them demonstrating very promising results in solving complex real world problems.

An evolving spiking neural network (eSNN) architecture was proposed in [18]. The eSNN belongs to the family of Evolving Connectionist Systems (ECoS), which was first introduced in [9]. ECoS based methods represent a class of constructive ANN algorithms that modify both the structure and connection weights of the network as part of the training process. Due to the evolving nature of the network and the employed fast one-pass learning algorithm, the method is able to accumulate information as it becomes available, without the requirement of retraining the network with previously

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 160–168, 2011.
© Springer-Verlag Berlin Heidelberg 2011

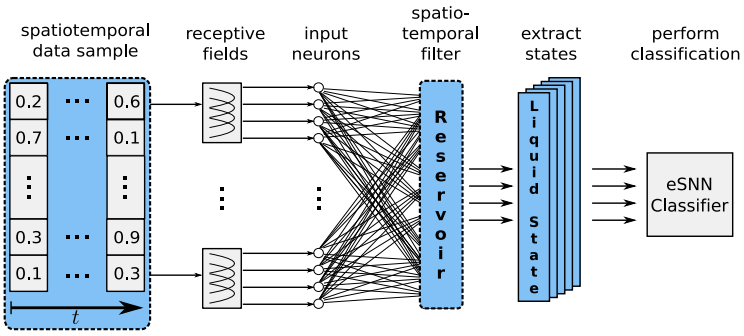


Fig. 1. Architecture of the extended eSNN capable of processing spatio-temporal data. The colored (dashed) boxes indicate novel parts in the original eSNN architecture.

presented data. The review in [17] summarises the latest developments on ECoS related research; we refer to [13] for a comprehensive discussion of the eSNN classification method.

The eSNN classifier learns the mapping from a single data vector to a specified class label. It is mainly suitable for the classification of time-invariant data. However, many data volumes are continuously updated adding an additional time dimension to the data sets. In [14], the authors outlined an extension of eSNN to reSNN which principally enables the method to process spatio-temporal information. Following the principle of a Liquid State Machine (LSM) [10], the extension includes an additional layer into the network architecture, *i.e.* a recurrent SNN acting as a reservoir. The reservoir transforms a spatio-temporal input pattern into a single high-dimensional network state which in turn can be mapped into a desired class label by the one-pass learning algorithm of eSNN.

In this paper, the reSNN extension presented in [14] is implemented and its suitability as a classification method is analyzed in computer simulations. We use a well-known real-world data set, *i.e.* the LIBRAS sign language data set [2], in order to allow an independent comparison with related techniques. The goal of the study is to gain some general insights into the working of the reservoir based eSNN classification and to deliver a proof of concept of its feasibility.

2 Spatio-temporal Pattern Recognition with reSNN

The reSNN classification method is built upon a simplified integrate-and-fire neural model first introduced in [16] that mimics the information processing of the human eye. We refer to [13] for a comprehensive description and analysis of the method. The proposed reSNN is illustrated in Figure 1. The novel parts in the architecture are indicated by the highlighted boxes. We outline the working of the method by explaining the diagram from left to right.

Spatio-temporal data patterns are presented to the reSNN system in form of an ordered sequence of real-valued data vectors. In the first step, each real-value of a data

vector is transformed into a spike train using a population encoding. This encoding distributes a single input value to multiple neurons. Our implementation is based on arrays of receptive fields as described in [1]. Receptive fields allow the encoding of continuous values by using a collection of neurons with overlapping sensitivity profiles.

As a result of the encoding, input neurons spike at predefined times according to the presented data vectors. The input spike trains are then fed into a spatio-temporal filter which accumulates the temporal information of all input signals into a single high-dimensional intermediate liquid state. The filter is implemented in form of a liquid or a reservoir [10], *i.e.* a recurrent SNN, for which the eSNN acts as a readout function. The one-pass learning algorithm of eSNN is able to learn the mapping of the liquid state into a desired class label. The learning process successively creates a repository of trained output neurons during the presentation of training samples. For each training sample a new neuron is trained and then compared to the ones already stored in the repository of the same class. If a trained neuron is considered to be too similar (in terms of its weight vector) to the ones in the repository (according to a specified similarity threshold), the neuron will be merged with the most similar one. Otherwise the trained neuron is added to the repository as a new output neuron for this class. The merging is implemented as the (running) average of the connection weights, and the (running) average of the two firing threshold. Because of the incremental evolution of output neurons, it is possible to accumulate information and knowledge as they become available from the input data stream. Hence a trained network is able to learn new data and new classes without the need of re-training already learned samples. We refer to [13] for a more detailed description of the employed learning in eSNN.

2.1 Reservoir

The reservoir is constructed of Leaky Integrate-and-Fire (LIF) neurons with exponential synaptic currents. This neural model is based on the idea of an electrical circuit containing a capacitor with capacitance C and a resistor with a resistance R, where both C and R are assumed to be constant. The dynamics of a neuron i are then described by the following differential equations:

$$\tau_m \frac{du_i}{dt} = -u_i(t) + R\, I_i^{\text{syn}}(t) \tag{1}$$

$$\tau_s \frac{dI_i^{\text{syn}}}{dt} = -I_i^{\text{syn}}(t) \tag{2}$$

The constant $\tau_m = RC$ is called the membrane time constant of the neuron. Whenever the membrane potential u_i crosses a threshold ϑ from below, the neuron fires a spike and its potential is reset to a reset potential u_r. We use an exponential synaptic current I_i^{syn} for a neuron i modeled by Eq. 2 with τ_s being a synaptic time constant.

In our experiments we construct a liquid having a small-world inter-connectivity pattern as described in [10]. A recurrent SNN is generated by aligning 100 neurons in a three-dimensional grid of size $4 \times 5 \times 5$. Two neurons A and B in this grid are connected with a connection probability

$$P(A, B) = C \times e^{\frac{-d(A,B)}{\lambda^2}} \tag{3}$$

where $d(A, B)$ denotes the Euclidean distance between two neurons and λ corresponds to the density of connections which was set to $\lambda = 2$ in all simulations. Parameter C depends on the type of the neurons. We discriminate into excitatory (ex) and inhibitory (inh) neurons resulting in the following parameters for C: $C_{ex-ex} = 0.3$, $C_{ex-inh} = 0.2$, $C_{inh-ex} = 0.5$ and $C_{inh-inh} = 0.1$. The network contained 80% excitatory and 20% inhibitory neurons. The connections weights were randomly selected by a uniform distribution and scaled in the interval $[-8, 8]$nA. The neural parameters were set to $\tau_m = 30$ms, $\tau_s = 10$ms, $\vartheta = 5$mV, $u_r = 0$mV. Furthermore, a refractory period of 5ms and a synaptic transmission delay of 1ms was used.

Using this configuration, the recorded liquid states did not exhibit the undesired behavior of over-stratification and pathological synchrony – effects that are common for randomly generated liquids [11]. For the simulation of the reservoir we used the SNN simulator Brian [4].

3 Experiments

In order to investigate the suitability of the reservoir based eSNN classification method, we have studied its behavior on a spatio-temporal real-world data set. In the next sections, we present the LIBRAS sign-language data, explain the experimental setup and discuss the obtained results.

3.1 Data Set

LIBRAS is the acronym for **LI**ngua **BRA**sileira de **S**inais, which is the official Brazilian sign language. There are 15 hand movements (signs) in the dataset to be learned and classified. The movements are obtained from recorded video of four different people performing the movements in two sessions. In total 360 videos have been recorded, each video showing one movement lasting for about seven seconds. From the videos 45 frames uniformly distributed over the seven seconds have then been extracted. In each frame, the centroid pixels of the hand are used to determine the movement. All samples have been organized in ten sub-datasets, each representing a different classification scenario. More comprehensive details about the dataset can be found in [2]. The data can be obtained from the UCI machine learning repository.

In our experiment, we used Dataset 10 which contains the hand movements recorded from three different people. This dataset is balanced consisting of 270 videos with 18 samples for each of the 15 classes. An illustration of the dataset is given in Figure 2. The diagrams show a single sample of each class.

3.2 Setup

As described in Section 2, a population encoding has been applied to transform the data into spike trains. This method is characterized by the number of receptive fields used for the encoding along with the width β of the Gaussian receptive fields. After some initial experiments, we decided to use 30 receptive fields and a width of $\beta = 1.5$. More details of the method can be found in [1].

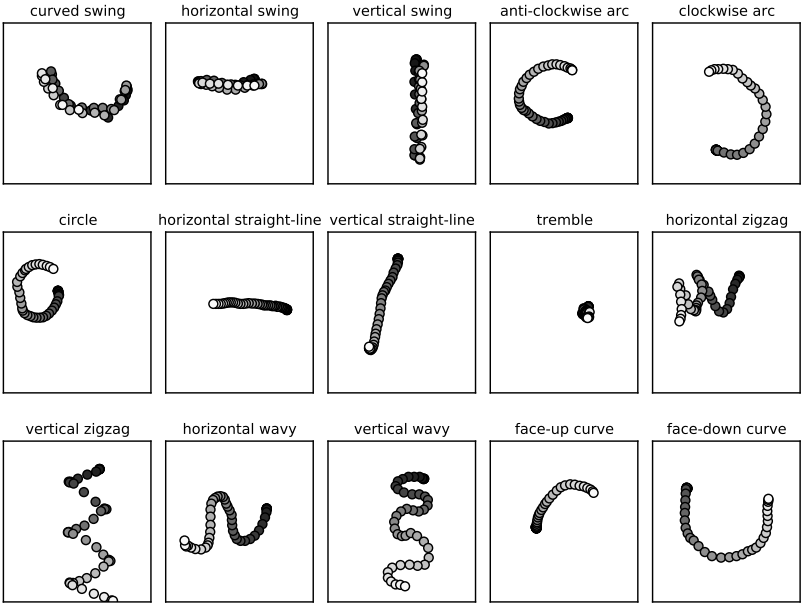


Fig. 2. The LIBRAS data set. A single sample for each of the 15 classes is shown, the color indicating the time frame of a given data point (black/white corresponds to earlier/later time points).

In order to perform a classification of the input sample, the state of the liquid at a given time t has to be read out from the reservoir. The way how such a liquid state is defined is critical for the working of the method. We investigate in this study three different types of readouts. We call the first type a cluster readout. The neurons in the reservoir are first grouped into clusters and then the population activity of the neurons belonging to the same cluster is determined. The population activity was defined in [3] and is the ratio of neurons being active in a given time interval $[t - \Delta_c t, t]$. Initial experiments suggested to use 25 clusters collected in a time window of $\Delta_c t = 10$ms. Since our reservoir contains 100 neurons simulated over a time period of $T = 300$ms, $T/\Delta_c t = 30$ readouts for a specific input data sample can be extracted, each of them corresponding to a single vector with 25 continuous elements. Similar readouts have also been employed in related studies [12].

The second readout is principally very similar to the first one. In the interval $[t - \Delta_f t, t]$ we determine the firing frequency of all neurons in the reservoir. According to our reservoir setup, this frequency readout produces a single vector with 100 continuous elements. We used a time window of $\Delta_f t = 30$ resulting in the extraction of $T/\Delta_f t = 10$ readouts for a specific input data sample.

Finally, in the analog readout, every spike is convolved by a kernel function that transforms the spike train of each neuron in the reservoir into a continuous analog signal. Many possibilities for such a kernel function exist, such as Gaussian and exponential kernels. In this study, we use the alpha kernel $\alpha(t) = e\ \tau^{-1}\ t\ e^{-t/\tau}\Theta(t)$ where $\Theta(t)$ refers to the Heaviside function and parameter $\tau = 10$ms is a time constant. The

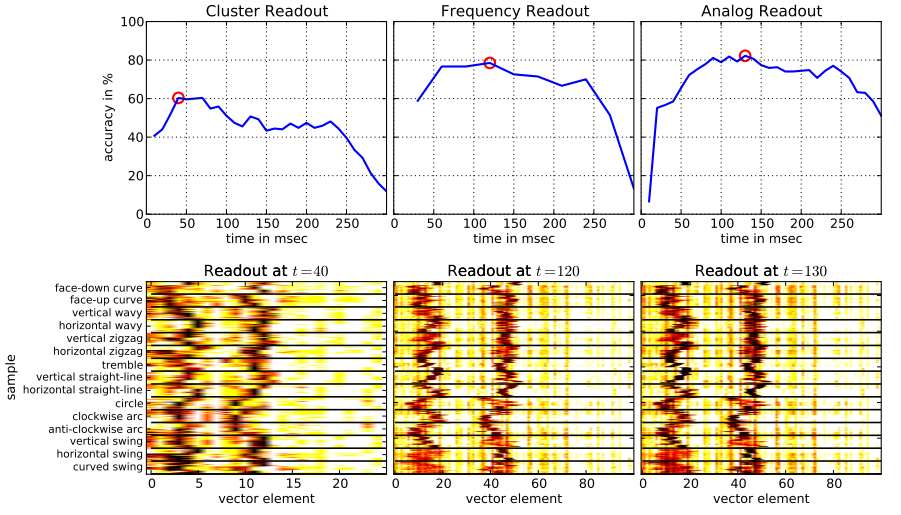


Fig. 3. Classification accuracy of eSNN for three readouts extracted at different times during the simulation of the reservoir (top row of diagrams). The best accuracy obtained is marked with a small (red) circle. For the marked time points, the readout of all 270 samples of the data are shown (bottom row).

convolved spike trains are then sampled using a time step of $\Delta_a t = 10$ms resulting in 100 time series – one for each neuron in the reservoir. In these series, the data points at time t represent the readout for the presented input sample. A very similar readout was used in [15] for a speech recognition problem. Due to the sampling interval Δ_a, $T/\Delta_a t = 30$ different readouts for a specific input data sample can be extracted during the simulation of the reservoir.

All readouts extracted at a given time have been fed to the standard eSNN for classification. Based on preliminary experiments, some initial eSNN parameters were chosen. We set the modulation factor $m = 0.99$, the proportion factor $c = 0.46$ and the similarity threshold $s = 0.01$. Using this setup we classified the extracted liquid states over all possible readout times.

3.3 Results

The evolution of the accuracy over time for each of the three readout methods is presented in Figure 3. Clearly, the cluster readout is the least suitable readout among the tested ones. The best accuracy found is 60.37% for the readout extracted at time 40ms, *cf.* the marked time point in the upper left diagram of the figure[1]. The readouts extracted at time 40ms are presented in the lower left diagram. A row in this diagram is the readout vector of one of the 270 samples, the color indicating the real value of the elements in that vector. The samples are ordered to allow a visual discrimination of the 15 classes. The first 18 rows belong to class 1 (curved swing), the next 18 rows to

[1] We note that the average accuracy of a random classifier is around $\frac{1}{15} \approx 6.67\%$.

class 2 (horizontal swing) and so on. Given the extracted readout vector, it is possible to even visually distinguish between certain classes of samples. However, there are also significant similarities between classes of readout vectors visible which clearly have a negative impact on the classification accuracy.

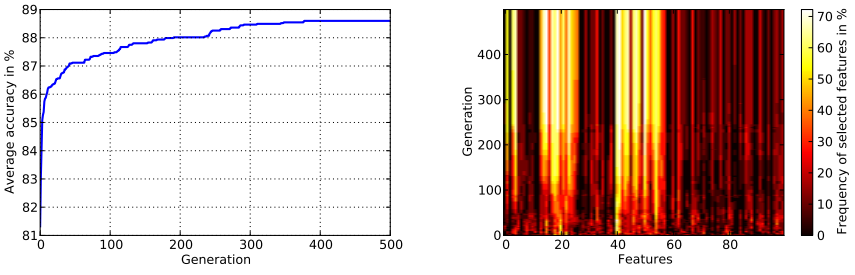
The situation improves when the frequency readout is used resulting in a maximum classification accuracy of 78.51% for the readout vector extracted at time 120ms, *cf.* middle top diagram in Figure 3. We also note the visibly better discrimination ability of the classes of readout vectors in the middle lower diagram: The intra-class distance between samples belonging to the same class is small, but inter-class distance between samples of other classes is large. However, the best accuracy was achieved using the analog readout extracted at time 130ms (right diagrams in Figure 3). Patterns of different classes are clearly distinguishable in the readout vectors resulting in a good classification accuracy of 82.22%.

3.4 Parameter and Feature Optimization of reSNN

The previous section already demonstrated that many parameters of the reSNN need to be optimized in order to achieve satisfactory results (the results shown in Figure 3 are as good as the suitability of the chosen parameters is). Here, in order to further improve the classification accuracy of the analog readout vector classification, we have optimized the parameters of the eSNN classifier along with the input features (the vector elements that represent the state of the reservoir) using the Dynamic Quantum inspired Particle swarm optimization (DQiPSO) [5]. The readout vectors are extracted at time 130ms, since this time point has reported the most promising classification accuracy. For the DQiPSO, 20 particles were used, consisting of eight update, three filter, three random, three embed-in and three embed-out particles. Parameter c_1 and c_2 which control the exploration corresponding to the global best (gbest) and the personal best (pbest) respectively, were both set to 0.05. The inertia weight was set to $w = 2$. See [5] for further details on these parameters and the working of DQiPSO. We used 18-fold cross validations and results were averaged in 500 iterations in order to estimate the classification accuracy of the model.

The evolution of the accuracy obtained from the global best particle during the PSO optimization process is presented in Figure 4a. The optimization clearly improves the classification abilities of eSNN. After the DQiPSO optimization an accuracy of 88.59% ($\pm 2.34\%$) is achieved. In comparison to our previous experiments [6] on that dataset, the time delay eSNN performs very similarly reporting an accuracy of 88.15% ($\pm 6.26\%$). The test accuracy of an MLP under the same conditions of training and testing was found to be 82.96% ($\pm 5.39\%$).

Figure 4b presents the evolution of the selected features during the optimization process. The color of a point in this diagram reflects how often a specific feature was selected at a certain generation. The lighter the color the more often the corresponding feature was selected at the given generation. It can clearly be seen that a large number of features have been discarded during the evolutionary process. The pattern of relevant features matches the elements of the readout vector having larger values, *cf.* the dark points in Figure 3 and compare to the selected features in Figure 4.



(a) Evolution of classification accuracy (b) Evolution of feature subsets

Fig. 4. Evolution of the accuracy and the feature subsets based on the global best solution during the optimization with DQiPSO

4 Conclusion and Future Directions

This study has proposed an extension of the eSNN architecture, called reSNN, that enables the method to process spatio-temporal data. Using a reservoir computing approach, a spatio-temporal signal is projected into a single high-dimensional network state that can be learned by the eSNN training algorithm. We conclude from the experimental analysis that the suitable setup of the reservoir is not an easy task and future studies should identify ways to automate or simplify that procedure. However, once the reservoir is configured properly, the eSNN is shown to be an efficient classifier of the liquid states extracted from the reservoir. Satisfying classification results could be achieved that compare well with related machine learning techniques applied to the same data set in previous studies. Future directions include the development of new learning algorithms for the reservoir of the reSNN and the application of the method on other spatio-temporal real-world problems such as video or audio pattern recognition tasks. Furthermore, we intend to develop a implementation on specialised SNN hardware [7,8] to allow the classification of spatio-temporal data streams in real time.

Acknowledgements. The work on this paper has been supported by the Knowledge Engineering and Discovery Research Institute (KEDRI, www.kedri.info). One of the authors, NK, has been supported by a Marie Curie International Incoming Fellowship with the FP7 European Framework Programme under the project “EvoSpike”, hosted by the Neuromorphic Cognitive Systems Group of the Institute for Neuroinformatics of the ETH and the University of Zürich.

References

1. Bohte, S.M., Kok, J.N., Poutré, J.A.L.: Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing 48(1-4), 17–37 (2002)
2. Dias, D., Madeo, R., Rocha, T., Biscaro, H., Peres, S.: Hand movement recognition for brazilian sign language: A study using distance-based neural networks. In: International Joint Conference on Neural Networks IJCNN 2009, pp. 697–704 (2009)

3. Gerstner, W., Kistler, W.M.: Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge (2002)
4. Goodman, D., Brette, R.: Brian: a simulator for spiking neural networks in python. BMC Neuroscience 9(Suppl 1), 92 (2008)
5. Hamed, H., Kasabov, N., Shamsuddin, S.: Probabilistic evolving spiking neural network optimization using dynamic quantum-inspired particle swarm optimization. Australian Journal of Intelligent Information Processing Systems 11(01), 23–28 (2010)
6. Hamed, H., Kasabov, N., Shamsuddin, S., Widiputra, H., Dhoble, K.: An extended evolving spiking neural network model for spatio-temporal pattern classification. In: 2011 International Joint Conference on Neural Networks, pp. 2653–2656 (2011)
7. Indiveri, G., Chicca, E., Douglas, R.: Artificial cognitive systems: From VLSI networks of spiking neurons to neuromorphic cognition. Cognitive Computation 1, 119–127 (2009)
8. Indiveri, G., Stefanini, F., Chicca, E.: Spike-based learning with a generalized integrate and fire silicon neuron. In: International Symposium on Circuits and Systems, ISCAS 2010, pp. 1951–1954. IEEE (2010)
9. Kasabov, N.: The ECOS framework and the ECO learning method for evolving connectionist systems. JACIII 2(6), 195–202 (1998)
10. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation 14(11), 2531–2560 (2002)
11. Norton, D., Ventura, D.: Preparing more effective liquid state machines using hebbian learning. In: International Joint Conference on Neural Networks, IJCNN 2006, pp. 4243–4248. IEEE, Vancouver (2006)
12. Norton, D., Ventura, D.: Improving liquid state machines through iterative refinement of the reservoir. Neurocomputing 73(16-18), 2893–2904 (2010)
13. Schliebs, S., Defoin-Platel, M., Worner, S., Kasabov, N.: Integrated feature and parameter optimization for an evolving spiking neural network: Exploring heterogeneous probabilistic models. Neural Networks 22(5-6), 623–632 (2009)
14. Schliebs, S., Nuntalid, N., Kasabov, N.: Towards Spatio-Temporal Pattern Recognition Using Evolving Spiking Neural Networks. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) ICONIP 2010, Part I. LNCS, vol. 6443, pp. 163–170. Springer, Heidelberg (2010)
15. Schrauwen, B., D'Haene, M., Verstraeten, D., Campenhout, J.V.: Compact hardware liquid state machines on fpga for real-time speech recognition. Neural Networks 21(2-3), 511–523 (2008)
16. Thorpe, S.J.: How can the human visual system process a natural scene in under 150ms? On the role of asynchronous spike propagation. In: ESANN. D-Facto public (1997)
17. Watts, M.: A decade of Kasabov's evolving connectionist systems: A review. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 39(3), 253–269 (2009)
18. Wysoski, S.G., Benuskova, L., Kasabov, N.K.: Adaptive Learning Procedure for a Network of Spiking Neurons and Visual Pattern Recognition. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2006. LNCS, vol. 4179, pp. 1133–1142. Springer, Heidelberg (2006)

An Adaptive Approach to Chinese Semantic Advertising

Jin-Yuan Chen, Hai-Tao Zheng*, Yong Jiang, and Shu-Tao Xia

Tsinghua-Southampton Web Science Laboratory, Graduate School at Shenzhen,
Tsinghua University, China
chenjinyuan87@gmail.com,
{zheng.haitao,jiangy,xiast}@sz.tsinghua.edu.cn

Abstract. Semantic Advertising is a new kind of web advertising to find the most related advertisements for web pages semantically. In this way, users are more likely to be interest in the related advertisements when browsing the web pages. A big challenge for semantic advertising is to match advertisements and web pages in a conceptual level. Especially, there are few studies proposed for Chinese semantic advertising. To address this issue, we proposed an adaptive method to construct an ontology automatically for matching Chinese advertisements and web pages semantically. Seven distance functions are exploited to measure the similarity between advertisements and web pages. Based on the empirical experiments, we found the proposed method shows a promising result in terms of precision, and among the distance functions, the Tanimoto distance function outperforms the other six distance functions.

Keywords: Semantic advertising, Chinese, Ontology, Distance function.

1 Introduction

With the development of the World Wide Web, advertising on the web is getting more and more important for companies. However, although users can see advertisements everywhere on the web, these advertisements on web pages may not attract users' attention, or even make them boring. Previous research [1] has shown that the more the advertisement is related to the page on which it displays, the more likely users will be interested on the advertisement and click it. Sponsored Search (SS) [2] and Contextual Advertising (CA) [3],[4],[5],[6],[7],[8],[9] are the two main methods to display related advertisements on web pages.

A main challenge for CA is to match advertisements and web pages based on semantics. Given a web page, it is hard to find an advertisement which is related to the web page on a conceptual level. Although A. Broder [3] has presented a method for match web pages and advertisements semantically using a taxonomic tree, the taxonomic tree is constructed by human experts, which costs much human effort and time-consuming. In addition, as the Chinese is different from English, semantic advertising based on Chinese is still very difficult. There are few methods proposed to address the Chinese semantic advertising. In the study, we focus on processing web pages and advertisements in Chinese. Especially, we develop an algorithm to

* Corresponding author.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 169–176, 2011.
© Springer-Verlag Berlin Heidelberg 2011

construct an ontology automatically. Based on the ontology, our method utilizes various distance functions to measure the similarities between web pages and advertisements. Finally, the proposed method is able to match web pages and advertisements on a conceptual level.

In summary, our main contributions are listed as follows:

1. A systematic method is proposed to process Chinese semantic advertising.
2. Developing an algorithm to construct the ontology automatically for semantic advertising.
3. Seven distance functions are utilized to measure the similarities between web pages and advertisements based on the constructed ontology. We have found that the taminoto distance has best performance for Chinese semantic advertising.

The paper proceeds as follows. In the next section, we review the related works in the web advertising domain. Section 3 articulates the Chinese semantic advertising architecture. Section 5 shows the experiment results for evaluation. The final section presents the conclusion and future work.

2 Related Work

In 2002, C.-N. Wang's research [1] presented that the advertisements in the pages should be relevant to the user's interest to avoid degrading the user's experience and increase the probability of reaction. In 2005 B. Ribeiro-Neto [4] proposes a method for contextual advertising. They use Bayesian network to generate a redefined document vector, and so the vocabulary impedance between web page and advertisement is much smaller. This network is composed by the k nearest documents (using traditional bag-of-word model), the target page or advertisement and all the terms in the k+1 documents. For each term in the network, the weight of the term is. $\rho\left((1-\alpha)\omega_{i0}+\alpha\sum_{j=1}^{k}\omega_{ij}sim\left(d_0,d_j\right)\right)$ In this way the document vector is extended to k+1 documents, and the system is able to find more related ads with a simple cosine similarity. M. Ciaramita [8] and T.-K. Fan [9] also solved this vocabulary impedance but using different hypothesis. In 2007, A. Broder [3] makes a semantic approach to contextual advertising. They classify both the ads and the target page into a big taxonomic tree. The final score of the advertisement is the combination of the TaxScore and the vector distance score. A. Anagnostopoulos [7] tested the contribution of different page parts for the match result based on this model. After that, Vanessa Murdock [5] uses statistical machine translation models to match ad and pages. They treat the vocabulary used in pages and ads as different language and then using translation methods to determine the relativity between the ad and page. Tao Mei [6] proposed a method that not just simply displays the ad in the place provided by the page, but displays in the image of the page.

3 Chinese Semantic Advertising Architecture

Semantic advertising is a process to advertise based on the context of the current page with a third-part ontology. The whole architecture is described in Figure 1.

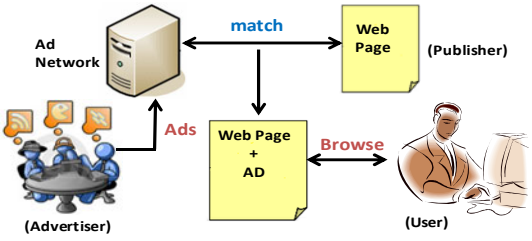


Fig. 1. The semantic advertising architecture

As discussed in [3], the main idea is to classify both page and advertisement to one or more concepts in ontology. With this classification information the algorithm calculates a score between the page and advertisement. This idea of the algorithm is described below:

```
(1) GetDocumentVector(page/advertisement d)
   return the top n terms and their tf-idf weight as a vector
(2) Classify(page/advertisement d)
   vector dv = GetDocumentVector(d)
   foreach(concept c in the ontology)
      vector cv = tf-idf of all the related phrases in c
     double score = distancemethod(cv,dv)
      put cv, score into the result vector
   return filtered concepts and their weight in the vector
(3) CalculateScore(page p, advertisement ad)
   vector pv = GetDocumentVector(p), av= GetDocumentVector(ad)
   vector pc= Classify(p), ac = Classify(ad)
   double ontoScore = conceptdistance(pc,ac)[3]
   double termScore = cosinedistance(pv,av)
   return ontoScore * alpha + (1-alpha) * termScore
```

There are still some problems need to be solved, they are listed below:

1. How to process Chinese web pages and advertisements?
2. How to build a comprehensive ontology for semantic advertising?
3. How to generate the related phrases for the ontology?
4. Which distance function is the best for similarity measurement?

The problems and corresponding solution are discussed in the following sections.

3.1 Preprocessing Chinese Web Pages and Advertisements

As Chinese articles do not contain blank chars between words, the first step to process a Chinese document must be word segmentation. We found a package called ICTCLAS [10] (Institute of Computing Technology, Chinese Lexical Analysis System) to solve this problem. This algorithm is developed by the Institute of Computing Technology, Chinese Academy of Science. Evaluation on ICTCLAS shows that its performance is competitive Compared with other systems: ICTCLAS has ranked top both in CTB and PK closed track. In PK open track, it ranks second position [11]. D. Yin [12], Y.-Q. Xia [13] and some other researchers use this system to finish their work.

The output format of this system is ({word}/{part of speech})+. For example, the result of "大家好" ("hello everyone") is "大家/rr 好/a", separated by blank space. In this result there are two words in the sentence, the first one is "大家" and the second one is "好". The parts of speech of them are "rr" and "a" meaning "personal pronoun" and "adjective". For more detailed document, please refer to [10]. Based on this result, we only process nouns and "Character Strings" in our algorithm because the words with other part of speech usually have little meaning. "Character String" is the word that combined by pure English characters and Arabic numerals, for example, "NBA", "ATP"," WTA2010" etc. And also, we build a stop list to filter some common words.

Besides that, the system maintains a dictionary for the names of the concepts in the ontology. All the words start with these words is translated to the class name. For example, "羽毛球拍"(Badminton racket) is one word in Chinese while "羽毛球"(Badminton) is a class name, then "羽毛球拍" is translated to "羽毛球".

3.2 The Ontology

Ontology is a formal explicit description of concepts in a domain of discourse [14], we build an ontology to describe the topics of web pages and advertisements. The ontology is also used to classify advertisements and pages based on the related phrases in its concepts. In a real system, there must be a huge ontology to match all the advertisements and pages. But for test, we build a small ontology focus on sports. The structure of the ontology is extracted from the trading platform in China called TaoBao [15], which is the biggest online trading platform in China. There are totally 25 concepts in the first level, and five of them have second level concepts. The average size of second level concepts is about ten. Figure 2 shows the ontology we used in our system.



Fig. 2. The ontology (Left side is the Chinese version and right side English)

3.3 Extracting Related Phrases for Ontology

Related phrases are used to match web pages and advertisements in a conceptual level. These phrases must be highly relevant to the class, and help the system to decide if the target document is related to this class. A. Broder [3] suggested that for each class about a hundred related phrases should be added. The system then calculates a centroid for each class which is used to measure the distance to the ad or page. But to build such ontology, it may cost several person years. Another problem is the imagination of one person is limited, he or she cannot add all the needed words into the system even with the help of some suggestion tools.

In our experiment, we develop another method using training method. We first select a number of web pages for training. For each page, we align it to a suitable concept in the constructed ontology manually (the page witch matches with more than one concept is filtered). Based on the alignment results, our method extracts ten keywords from each web page and treats them as a related phrase of the aligned concept. The keyword extraction algorithm is the traditional TF-IDF method. Consequently, each concept in the constructed ontology has a group of related phrases.

3.4 The Distance Function

In this paper, we utilize seven distance functions to measure the similarity between web pages or advertisements with the ontology concepts. Assuming that $\boldsymbol{c}=(c_1,\ldots,c_m)$, $\boldsymbol{c}'=(c'_1,\ldots,c'_m)$ are the two term vector, the weight of each term is the tf-idf value of it, these seven distance are:

• Euclidean distance:
$$d_{EUC}(\boldsymbol{c},\boldsymbol{c}')=\sqrt{\sum_{i=1}^{m}(c_i-c'_i)^2} \tag{1}$$

• Canberra distance:
$$d_{CAN}(\boldsymbol{c},\boldsymbol{c}')=\sum_{i=1}^{m}\frac{|c_i-c'_i|}{|c_i|+|c'_i|} \tag{2}$$

When divide by zero occurs, this distance is defined as zero. In our experiment, this distance may be very close to the dimension of the vectors (For most cases, there are only a small number of words in a concept's related phrases also appears in the page). In this situation the concepts with more related phrases tend to be further even if they are the right class. Finally we use $1/(dimension-d_{CAN})$ for this distance.

• Cosine distance
$$d_{COS}(\boldsymbol{c},\boldsymbol{c}')=\frac{\sum_{i=1}^{m}(c_i*c'_i)}{|c|*|c'|} \tag{3}$$

• Chebyshev distance:
$$d_{EUC}(\boldsymbol{c},\boldsymbol{c}')=\max_{1\le i\le m}|c_i-c'_i| \tag{4}$$

• Hamming distance:
$$d_{HAM}(\boldsymbol{c},\boldsymbol{c}')=\sum_{i=1}^{m}isDiff(c_i,c'_i) \tag{5}$$

Where $isDiff(c_i,c'_i)$ is 1 if c_i and c'_i are different, and 0 if they're equal. As same as Canberra distance, we finally use $1/(dimension-d_{HAM})$ for this distance.

• Manhattan distance: $$d_{MAN}(\boldsymbol{c},\boldsymbol{c}') = \sum_{i=1}^{m} |c_i - c_i'| \quad (6)$$

• Tanimoto distance: $$d_{TAN}(\boldsymbol{c},\boldsymbol{c}') = \frac{\sum_{i=1}^{m} (c_i * c_i')}{|\boldsymbol{c}|^2 + |\boldsymbol{c}'|^2 - \sum_{i=1}^{m} (c_i * c_i')} \quad (7)$$

The definitions of the first six distances are from V. Martinez's work [16]. And the definition of Tanimoto distance can be found in [17], the WikiPedia.

4 Evaluation

4.1 Experiment Setup

To test the algorithm, we find 400 pages and 500 ads in the area sport. And then we choose 200 as training set, the other 200 as the test set. The pages in the test set are mapped to a number of related ads artificially, while the pages in the training set have its ontology information. A simple result trained by all the pages in the training set is not enough, we also need to know the training result with different training set size (from 0 to 200). In order to ensure all the classes have the similar size of training pages, we iterator over all the classes and randomly select one unused page that belongs to this class for training until the total page selected reaches the expected size. To make sure there is no bias while choosing the pages, for each training size, we run our experiment for max(200/size + 1, 10) times, the final result is the average of the experiments. We use the precision measurement in our experiment because users only care about the relevance between the advertisement and the page:

$$Precision(n) = \frac{The\ number\ of\ \ relevant\ ads\ in\ the\ \ first\ n\ results}{n} \quad (8)$$

4.2 Experiment Results

In order to find out the best distance function, we draw Figure 3 to compare the results. The values of each method in the figure are the average number of the results with different training set size.

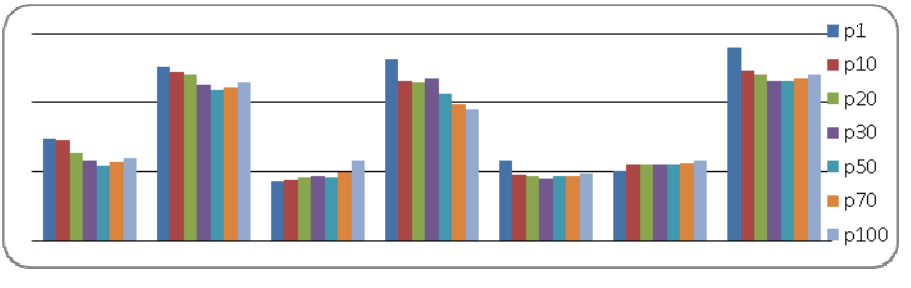


Fig. 3. The average precision of the seven distance functions

From Figure 3, we found that Canberra, Cosine and Tanimoto perform much better than the other four methods. Averagely, precisions for the three methods are Canberra 59%, cosine 58% and Tanimoto 65%. The precision of cosine similarity is much lower than Canberra and Tanimoto in P70 and P80. We conclude that Canberra distance and Tanimoto distance is better than cosine distance. In order to find out which of the two methods is better, we draw the detailed training result view. Figure 4 shows the training result of these two methods.

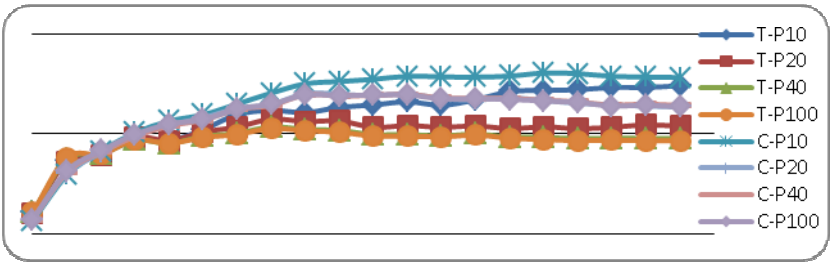


Fig. 4. The training result, C refers to Canberra, and T for Tanimoto

From Figure 4, we find that the maximum precision of Tanimoto and Canberra are almost the same (80% for P10 and 65%for others) while Tanimoto is a litter higher than Canberra. The training result shows that the performance falls down obviously while training set size reaches 80 for Canberra distance. This phenomenon is not suitable for our system, as a concept is expected to have about 100 related phrases, while a training size 80 means about ten related phrases for each class. And for Tanimoto distance, the performance falls only a little while training size increases. From these analyze, we conclude that the tanimoto distance is best for our system.

5 Conclusion and Future Work

In this paper, we proposed a semantic advertising method for Chinese. Focusing on processing web pages and advertisements in Chinese, we develop an algorithm to automatically construct an ontology. Based on the ontology, our method exploits seven distance functions to measure the similarities between web pages and advertisements. A main difference between Chinese and English processing is that Chinese documents needs to be segmented into words first, which contributes a big influence to the final matching results. The empirical experiment results indicate that our method is able to match web pages and advertisements with a relative high precision (80%). Among the seven distance functions, Tanimoto distance shows best performance.

In the future, we will focus on the optimization of the distance algorithm and the training method. For the distance algorithm, there still remains some problem. That is a node with especially huge related phrases will seems further than a smaller one. As the related phrases increases, it is harder to separate the right classes from noisy classes, because the distances of these classes are all very big. For training algorithm,

we need to optimize the extraction method for related phrases by using a better keyword extraction method, such as [18], [19], and [20].

Acknowledgments. This research is supported by National Natural Science Foundation of China (Grant No. 61003100) and Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20100002120018).

Reference

1. Wang, C.-N., Zhang, P., Choi, R., Eredita, M.D.: Understanding consumers attitude toward advertising. In: Eighth Americas Conference on Information System, pp. 1143–1148 (2002)
2. Fain, D., Pedersen, J.: Sponsored search: A brief history. In: Proc. of the Second Workshop on Sponsored Search Auctions, 2006. Web publication (2006)
3. Broder, A., Fontoura, M., Josifovski, V., Riedel, L.: A semantic approach to contextual advertising. In: SIGIR 2007. ACM Press (2007)
4. Ribeiro-Neto, B., Cristo, M., Golgher, P.B., de Moura, E.S.: Impedance coupling in content-targeted advertising. In: SIGIR 2005, pp. 496–503. ACM Press (2005)
5. Murdock, V., Ciaramita, M., Plachouras, V.: A Noisy-Channel Approach to Contextual Advertising. In: ADKDD 2007 (2007)
6. Mei, T., Hua, X.-S., Li, S.-P.: Contextual In-Image Advertising. In: MM 2008 (2008)
7. Anagnostopoulos, A., Broder, A.Z., Gabrilovich, E., Josifovski, V., Riedel, L.: Just-in-Time Contextual Advertising. In: CIKM 2007 (2007)
8. Ciaramita, M., Murdock, V., Plachouras, V.: Semantic Associations for Contextual Advertising. Journal of Electronic Commerce Research 9(1) (2008)
9. Fan, T.-K., Chang, C.-H.: Sentiment-oriented contextual advertising. Knowledge and Information Systems (2010)
10. The ICTCLAS Web Site, `http://www.ictclas.org`
11. Zhang, H.-P., Yu, H.-K., Xiong, D.Y., Liu, Q.: HHMM-based Chinese lexical analyzer ICTCLAS. In: SIGHAN 2003, Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, vol. 17 (2003)
12. Yin, D., Shao, M., Jiang, P.-L., Ren, F.-J., Kuroiwa, S.: Treatment of Quantifiers in Chinese-Japanese Machine Translation. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCS (LNAI), vol. 4114, pp. 930–935. Springer, Heidelberg (2006)
13. Xia, Y.-Q., Wong, K.-F., Gao, W.: NIL Is Not Nothing: Recognition of Chinese Network Informal Language Expressions. In: 4th SIGHAN Workshop at IJCNLP 2005 (2005)
14. Noy, N.F., McGuinness, D.L.: Ontology development 101: A guide to creating your first ontology. Technical Report SMI-2001-0880, Stanford Medical Informatics (2001)
15. TaoBao, `http://www.taobao.com`
16. Martinez, V., Simari, G.I., Sliva, A., Subrahmanian, V.S.: Convex: Similarity-Based Algorithms for Forecasting Group Behavior. IEEE Intelligent Systems 23, 51–57 (2008)
17. Jaccard index, `http://en.wikipedia.org/wiki/Jaccard_index`
18. Yih, W.-T., Goodman, J., Carvalho, V.R.: Finding Advertising Keywords on Web Pages. In: WWW (2006)
19. Zhang, C.-Z.: Automatic Keyword Extraction from Documents Using Conditional Random Fields. Journal of Computational Information Systems (2008)
20. Chien, L.F.: PAT-tree-based keyword extraction for Chinese information retrieva. In: SIGIR 1997. ACM, New York (1997)

A Lightweight Ontology Learning Method for Chinese Government Documents

Xing Zhao, Hai-Tao Zheng*, Yong Jiang, and Shu-Tao Xia

Tsinghua-Southampton Web Science Laboratory, Graduate School at Shenzhen,
Tsinghua University. 518055 Shenzhen, P.R. China
zhaoxing05@gmail.com,
{zheng.haitao,jiangy,xiast}@sz.tsinghua.edu.cn

Abstract. Ontology learning is a way to extract structure data from natural documents. Recently, Data-government is becoming a new trend for governments to open their data as linked data. However, there are few methods proposed to generate linked data based on Chinese government documents. To address this issue, we propose a lightweight ontology learning approach for Chinese government documents. Our method automatically extracts linked data from Chinese government documents that consist of government rules. Regular Expression is utilized to discover the semantic relationship between concepts. This is a lightweight ontology learning approach, though cheap and simple, it is proved in our experiment that it has a relative high precision value (average 85%) and a relative good recall value (average 75.7%).

Keywords: Ontology Learning, Chinese government documents, Semantic Web.

1 Introduction

Recent years, with the development of E-Government [1], governments begin to publish information onto the web, in order to improve transparency and interactivity with citizens. However, most governments now just provide simple search tools such as keyword search to the citizens. Since there is huge number of government documents covering almost every area of the life, keyword search often returns great number of results. Looking though all the results to find appropriate result is actually a tedious task.

Data-government [2] [3], which uses Semantic Web technologies, aims to provide a linked government data sharing platform. It is based on linked-data, which is presented as the machine readable data formats instead of the original text format that can be only read by human. It provides powerful semantic search, with that citizens can easily find what concepts they need and the relationship of the concepts.

However, before we use linked-data to provide semantic search functions, we need to generate linked data from documents. Most of the existing techniques for ontology learning from text require human effort to complete one or more steps of the whole

* Corresponding author.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 177–184, 2011.
© Springer-Verlag Berlin Heidelberg 2011

process. For Chinese documents, since NLP (Nature Language Process) for Chinese is much more difficult than English, automatic ontology learning from Chinese text presents a great challenge.

To address this issue, we present an unsupervised approach that automatically extracts linked data from Chinese government document which consists of government rules. The extraction approach is based on regular expression (Regex, in short) matching, and finally we use the extracted linked data to create RDF files. This is a lightweight ontology learning approach, though cheap and simple, it is proved in our experiment that it has a high precision rate (average 85%) and a good recall rate (average 75.7%).

The remaining sections in this paper are organized as follows. Section 2 discusses the related work of the ontology learning from text. We then introduce our approach fully in Section 3. In Section 4, we provide the evaluation methods and our experiment, with some briefly analysis. Finally, we make concluding remarks and discuss future work in Section 5.

2 Related Work

Existing approaches for ontology learning from structured data sources and semi-structured data sources have been proposed a lot and presented good results [4]. However, for unstructured data, such as text documents, web pages, there is little approach presenting good results in a completely automated fashion [5].

According to the main technique used for discovering knowledge relevant, traditional methods for ontology learning from texts can be grouped into three classes: Approaches based on linguistic techniques [6] [7]; Approaches based on statistical techniques [8] [9]; Approaches based on machine learning algorithms [10] [11]. Although some of these approaches present good results, human effort is necessary to complete one or more steps of the whole process in almost all of them.

Since it is much more difficult to do NLP with Chinese text than English text, there is little automatic approach to do ontology learning for Chinese text until recently. In [12], an ontology learning process that based on chi-square statistics is proposed for automatic learning an Ontology Graph from Chinese texts for different domains.

3 Ontology Learning for Chinese Government Documents

Most of the Chinese government documents are mainly composed of government rules and have the similar form like the one that Fig. 1 provides.

深圳经济特区出租屋管理若干规定

(Rules of Shenzhen Municipality on the Administration of Leasing Houses)

第一条　本规定所称出租屋，包括住宅出租屋和工商业出租屋及其他出租屋。

(**Article 1** The leasing houses mentioned in these rules include the leasing houses for residence, industry, commerce or other use.)

第二条　……

(**Article 2**……)

……

Fig. 1. An example of Chinese government document

Government rules are basic function unit of a government document. Fig. 2 shows an example of government rule.

第一条　本规定所称出租屋，包括住宅出租屋和工商业出租屋及其他出租屋。
(**Article 1** The leasing houses mentioned in these rules include the leasing houses for residence, industry, commerce or other use.)

Fig. 2. An example of government rule

The ontology learning steps of our approach include preprocess, term extraction, government rule classification, triple creation, and RDF generation.

3.1 Preprocess

Government Rule Extraction with Regular Expression. We extract government rules from the original documents with Regular Expression (Regex) [13] as pattern matching method. The Regex of the pattern of government rules is

$$第[一二三四五六七八九十]+条[\\s]+[^。]+。 \quad . \tag{1}$$

We traverse the whole document and find all government rules matching the Regex, then create a set of all government rules in the document.

Chinese Word Segmentation and Filtering. Compared to English, Chinese sentence is always without any blanks to segment words. We use ICTCLAS [14] as our Chinese lexical analyzer to segment Chinese text into words and tag each word with their part of speech. For instance, the government rule in Fig. 2 is segmented and tagged to words sequence in Fig. 3.

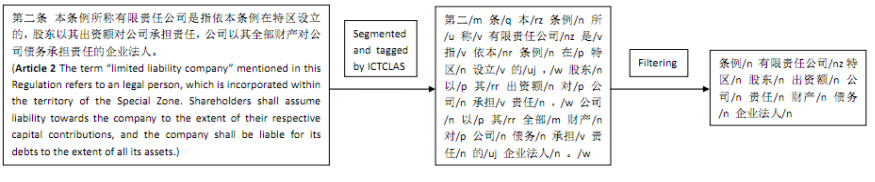


Fig. 3. Segmentation and Filtering

In this sequence, words are followed by their part of speech. For example, "有限责任公司/nz", where symbol "/nz" represents that word "有限责任公司"(limited liability company) is a proper noun.

According to our statistics, substantive words usually contain much more important information than other words in government rules. As Fig. 3 shows, after segmentation and tagging, we do a filtering to filter substantive words and remove duplicate words in a government rule.

By preprocessing, we convert original government documents into sets of government rules. For each government rule in the set, there is a related set of words. Each set holds the substantive words of the government rule.

3.2 Term Extraction

To extract key concept of government documents, we use TF-IDF measure to extract keywords from the substantive words set of each government rule.

For each document, we create a term set consists of the keywords, which represent the key concept of the document. The number of keywords extracted from each document will make great effect to the results and more discussion is in Section 4.

3.3 Government Rule Classification

In this step, we find out the relationship of key concept and government rules. According to our statistics, most of the Chinese government documents are mainly composed of three types of government rules:

Definition Rule. Definition Rule is a government rule which defines one or more concepts. Fig. 2 provides an example of Definition Rule.

According to our statistics, its most obvious signature is that it is a declarative sentence with one or more judgment word, such as “是”, “为” (It is approximately equal to “be” in English, but in Chinese, judgment word has very little grammatical function, almost only appears in declarative sentence).

Obligation Rule. Obligation Rule is a government rule which provides obligations. Fig. 4 provides an example of Obligation Rule.

第四条 各级人民政府应当加强对出租屋管理的领导和协调。
(**Article 4** The people's governments at all levels shall strengthen the leadership and coordination to the administration of leasing houses.)

Fig. 4. An example of Obligation Rule

According to our statistics, its most obvious signature is including one or more modal verb, such as “应当(shall)”, “必须(must)”, “不应(shall not)”.

Requirement Rule. Requirement Rule is a government rule which claims the requirement of government formalities. Fig. 5 provides an example of Requirment Rule.

第九十三条 公司财产拨付清算费用后，按下列顺序清偿：
（一）员工工资和社会保险费；（二）税款；（三）公司债务。
(**Article 93** The assets after paying of f the expenses of the company shall be liquidated in accordance with the following orders:
(1) The wages and social insurance premiums of the staff and workers of the company; (2) The taxation; (3) The debts of the company.)

Fig. 5. An example of Requirment Rule

According to our statistics, its most obvious signature is including one or more special words , such as “具备(have)”, “下列条件(following orders)”, following by a list of requirements.

We use Regex as our pattern matching approach to match the special signature of government rules in rule set.

For Definition Rule, the Regex is:

$$第[^条]+条\\s+（[^。]+term[^。]+（是|为）[^。]+。） \quad . \tag{2}$$

For Obligation Rule, it is:

$$第[^条]+条\\s+（[^。]+term[^。]+（应当|必须|不应）[^。]+。） \quad . \tag{3}$$

And for Requirement Rule, it is:

$$第[^条]+条\\s+([^。]+term\ [^。]+(具备|下列条件|（[^）]+）)[^。]+。) . \tag{4}$$

Where the “term” represents the term we extract from each document.

We traverse the whole government rule set created in Step 1; find all government rules with the given term and matching the Regex. Thus, we classify the government rule set into three classes, which includes definition rules, obligation rules, requirement rules separately.

3.4 Triple Creation

RDF graphs are made up of collections of triples. Triples are made up of a subject, a predicate, and an object. In Step 3 (rule classification), the relationship of key concept and government rules is established. To create triples, we traverse the whole government rule set and get term as subject, class as predicate, and content of the rule as object. For example, the triple of the government rule in Fig. 2 is shown in Fig. 6:



Fig. 6. Triple of the government rule

3.5 RDF Generation

We use Jena [15] to merge triples to a whole RDF graph and finally generate RDF files.

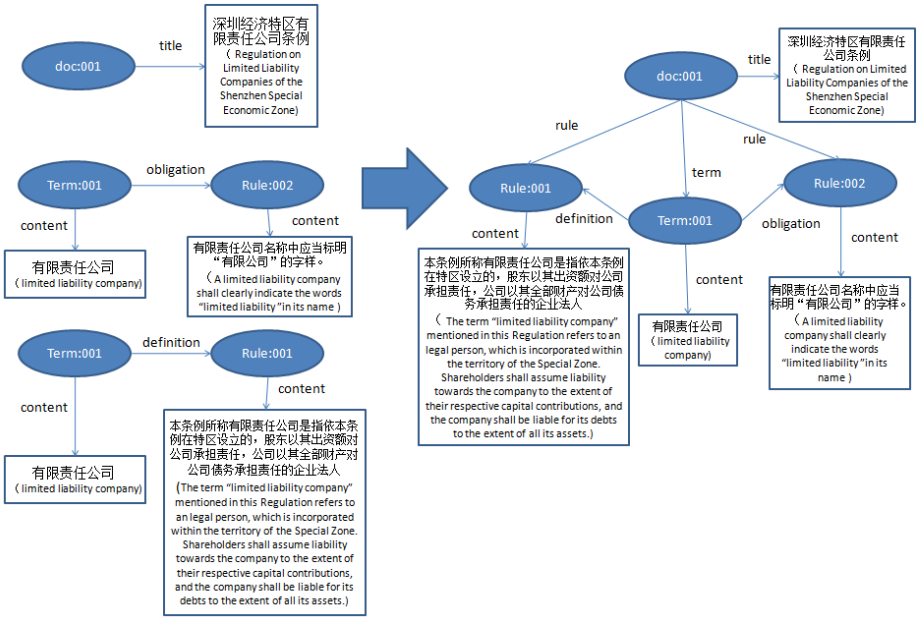


Fig. 7. RDF graph generation process

4 Evaluation

4.1 Experiment Setup

We use government documents from Shenzhen Nanshan Government Online [16] as data set. There are 302 government documents with about 15000 government rules. For evaluation, we random choose 41 of all the documents as test set, which contains 2010 government rules. We make two evaluation experiments to evaluate our method.

The first experiment aims at measuring the precision and recall of our method. The main steps of the experiment are as follows:

(a) Domain experts are requested to classify government rules in the test set, and tag them with “Definition Rule”, “Obligation Rule”, “Requirement Rule” and “Unknown Rule”. Thus, we get a benchmark. (b) We use our approach to process government rules in the same test set and compare results with the benchmark. Finally, we calculate precision and recall of our approach.

In Step 2(Term Extraction), we mention that the number of keywords extracted from a document will make great effect to the results. We make an experiment with different number of keywords (from 3 to 15), the results are provided in Fig. 8.

The second experiment compares semantic search with the linked data created by our approach to keyword search. Domain experts are asked to use two search methods to search same concepts. Then we analyze the precision of them. This experiment aims at evaluating the accuracy of the linked data. The results are provided in Fig. 9.

4.3 Results

Fig. 8 provides the precision and recall for different number of keywords. It is clear that more keywords yield high recall, but precision is almost no difference. When number of keywords is more than 10, there is little increase if we add more keywords. It is mainly because there are no related government rules with new added in keywords. The results also prove that our approach is trustable, with high precision (above 80%) whenever keywords set are small or large. And if we take enough keywords number (>10), recall will surpass 75%.

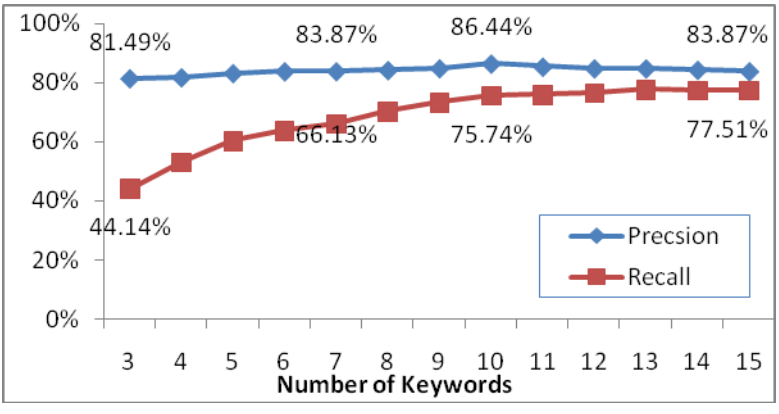


Fig. 8. Precision and Recall based on different number of keywords

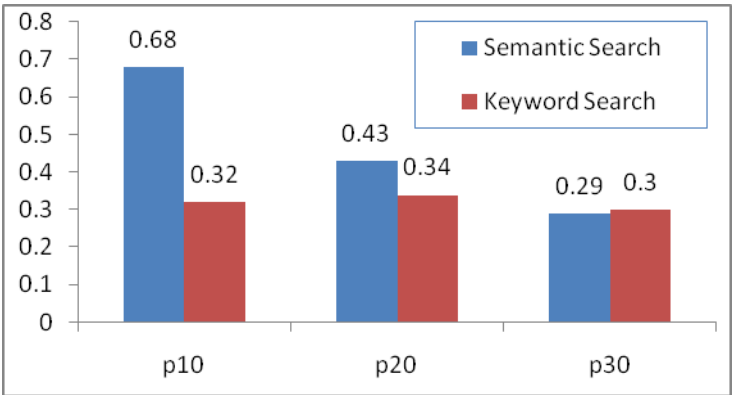


Fig. 9. Precision value for two search methods

Fig. 9 provides the precision value of different search methods, Semantic Search and Keyword Search. Keyword Search application is implemented based on Apache Lucene [17]. Linked data created by our approach provides good accuracy, for p10, that is 68%. This is very meaningful for users, since they often look though the first page of search results only.

5 Conclusion and Future Work

In this paper, a lightweight ontology learning approach is proposed for Chinese government document. The approach automatically extracts linked data from Chinese government document which consists of government rules. Experiment results demonstrate that it has a relatively high precision rate (average 85%) and a good recall rate (average 75.7%).

In future work, we will extract more types of relationship of the term and government rules. The concept extraction method may be changed in order to deal with multi-word concept.

Acknowledgments. This research is supported by National Natural Science Foundation of China (Grant No. 61003100 and No. 60972011) and Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20100002120018 and No. 2010000211033).

References

1. e-Government, http://en.wikipedia.org/wiki/E-Government
2. DATA.GOV, http://www.data.gov/
3. data.gov.uk, http://data.gov.uk/
4. Lehmann, J., Hitzler, P.: A Refinement Operator Based Learning Algorithm for the ALC Description Logic. In: Blockeel, H., Ramon, J., Shavlik, J., Tadepalli, P. (eds.) ILP 2007. LNCS (LNAI), vol. 4894, pp. 147–160. Springer, Heidelberg (2008)
5. Drumond, L., Girardi, R.: A survey of ontology learning procedures. In: WONTO 2008, pp. 13–25 (2008)
6. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: COLING 1992, pp. 539–545 (1992)
7. Hahn, U., Schnattinger, K.: Towards text knowledge engineering. In: AAAI/IAAI 1998, pp. 524–531. The MIT Press (1998)
8. Agirre, E., Ansa, O., Hovy, E.H., Martinez, D.: Enriching very large ontologies using the www. In: ECAI Workshop on Ontology Learning, pp. 26–31 (2000)
9. Faatz, A., Steinmetz, R.: Ontology enrichment with texts from the WWW. In: Semantic Web Mining, p. 20 (2002)
10. Hwang, C.H.: Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information. In: KRDB 1999, pp. 14–20 (1999)
11. Khan, L., Luo, F.: Ontology construction for information selection. In: ICTAI 2002, pp. 122–127 (2002)
12. Lim, E.H.Y., Liu, J.N.K., Lee, R.S.T.: Knowledge Seeker - Ontology Modelling for Information Search and Management. Intelligent Systems Reference Library, vol. 8, pp. 145–164. Springer, Heidelberg (2011)
13. Regular expression, http://en.wikipedia.org/wiki/Regular_expression
14. ICTCLAS, http://www.ictclas.org/
15. Jena, http://jena.sourceforge.net/
16. Nanshan Government Online, http://www.szns.gov.cn/
17. Apache Lucene, http://lucene.apache.org/

Relative Association Rules Based on Rough Set Theory

Shu-Hsien Liao[1], Yin-Ju Chen[2], and Shiu-Hwei Ho[3]

[1] Department of Management Sciences, Tamkang University,
No.151 Yingzhuan Rd., Danshui Dist., New Taipei City 25137, Taiwan R.O.C
[2] Graduate Institute of Management Sciences, Tamkang University,
No.151 Yingzhuan Rd., Danshui Dist., New Taipei City 25137, Taiwan R.O.C
[3] Department of Business Administration,
Technology and Science Institute of Northern Taiwan,
No. 2, Xueyuan Rd., Peitou, 112 Taipei, Taiwan, R.O.C
michael@mail.tku.edu.tw, s5515124@ms18.hinet.net,
succ04.dba@msa.hinet.net

Abstract. The traditional association rule that should be fixed in order to avoid the following: only trivial rules are retained and interesting rules are not discarded. In fact, the situations that use the relative comparison to express are more complete than those that use the absolute comparison. Through relative comparison, we proposes a new approach for mining association rule, which has the ability to handle uncertainty in the classing process, so that we can reduce information loss and enhance the result of data mining. In this paper, the new approach can be applied for finding association rules, which have the ability to handle uncertainty in the classing process, is suitable for interval data types, and help the decision to try to find the relative association rules within the ranking data.

Keywords: Rough set, Data mining, Relative association rule, Ordinal data.

1 Introduction

Many algorithms have been proposed for mining Boolean association rules. However, very little work has been done in mining quantitative association rules. Although we can transform quantitative attributes into Boolean attributes, this approach is not effective, is difficult to scale up for high-dimensional cases, and may also result in many imprecise association rules [2]. In addition, the rules express the relation between pairs of items and are defined in two measures: support and confidence. Most of the techniques used for finding association rule scan the whole data set, evaluate all possible rules, and retain only those rules that have support and confidence greater than thresholds. It's mean that the situations that use the absolute comparison [3]. The remainder of this paper is organized as follows. Section 2 reviews relevant literature in correlation with research and the problem statement. Section 3 incorporation of rough set for classification processing. Closing remarks and future work are presented in Section 4.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 185–192, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Literature Review and Problem Statement

In the traditional design, Likert Scale uses a checklist for answering and asks the subject to choose only one best answer for each item. The quantification of the data is equal intervals of integer. For example, age is the most common type for the quantification data that have to transform into an interval of integer. Table 1 and Table 2 present the same data. The difference is due to the decision maker's background. One can see that the same data of the results has changed after the decision maker transformation of the interval of integer. An alternative is the qualitative description of process states, for example by means of the discretization of continuous variable spaces in intervals [6].

Table 1. A decision maker

No	Age	Interval of integer
t_1	20	20–25
t_2	23	26–30
t_3	17	Under 20
t_4	30	26–30
t_5	22	20–25

Table 2. B decision maker

No	Age	Interval of integer
t_1	20	Under 25
t_2	23	Under 25
t_3	17	Under 25
t_4	30	Above 25
t_5	22	Under 25

Furthermore, in this research, we incorporate association rules with rough sets and promote a new point of view in applications. In fact, there is no rule for the choice of the "right" connective, so this choice is always arbitrary to some extent.

3 Incorporation of Rough Set for Classification Processing

The traditional association rule, which pays no attention to finding rules from ordinal data. Furthermore, in this research, we incorporate association rules with rough sets and promote a new point of view in interval data type applications. The data processing of interval scale data is described as below.

First: Data processing—Definition 1—Information system: Transform the questionnaire answers into information system $IS = (U, Q)$, where $U = \{x_1, x_2, \cdots x_n\}$ is a finite set of objects. Q is usually divided into two parts, $G = \{g_1, g_2, \cdots g_i\}$ is a finite set of general attributes/criteria, and $D = \{d_1, d_2, \cdots d_k\}$ is a set of decision attributes. $f_g = U \times G \to V_g$ is called the information function, V_g is the domain of the attribute/criterion g, and f_g is a total function such that $f(x, g) \in V_g$ for each $g \in Q$; $x \in U$. $f_d = U \times D \to V_d$ is called the sorting decision-making information function, V_d is the domain of the decision attributes/criterion d, and f_d is a total function such that $f(x, d) \in V_d$ for each $d \in Q$; $x \in U$.

Example: According to Tables 3 and 4, x_1 is a male who is thirty years old and has an income of 35,000. He ranks beer brands from one to eight as follows: Heineken,

Miller, Taiwan light beer, Taiwan beer, Taiwan draft beer, Tsingtao, Kirin, and Budweiser.

Then:

$$f_{d_1} = \{4,3,1\} \qquad f_{d_2} = \{4,3,2,1\} \qquad f_{d_3} = \{6,3\} \qquad f_{d_4} = \{7,2\}$$

Table 3. Information system

U \ Q	General attributes G		Decision-making D
	Item1: Age g_1	Item2: Income g_2	Item3: Beer brand recall
x_1	30 g_{1_1}	35,000 g_{2_1}	As shown in Table 4.
x_2	40 g_{1_2}	60,000 g_{2_2}	As shown in Table 4.
x_3	45 g_{1_3}	80,000 g_{2_4}	As shown in Table 4.
x_4	30 g_{1_1}	35,000 g_{2_1}	As shown in Table 4.
x_5	40 g_{1_2}	70,000 g_{2_3}	As shown in Table 4.

Table 4. Beer brand recall ranking table

U	D the sorting decision-making set of beer brand recall							
	Taiwan beer d_1	Heineken d_2	light beer d_3	Miller d_4	draft beer d_5	Tsingtao d_6	Kirin d_7	Budweiser d_8
x_1	4	1	3	2	5	6	7	8
x_2	1	2	3	7	5	6	4	8
x_3	1	4	3	2	5	6	7	8
x_4	3	1	6	2	5	4	8	7
x_5	1	3	6	2	5	4	8	7

Definition 2: The Information system is a quantity attribute, such as g_1 and g_2, in Table 3; therefore, between the two attributes will have a covariance, denoted by $\sigma_G = Cov(g_i, g_j)$. $\rho_G = \dfrac{\sigma_G}{\sqrt{Var(g_i)}\sqrt{Var(g_j)}}$ denote the population correlation coefficient and $-1 \le \rho_G \le 1$.

Then:

$$\rho_G^+ = \{g_{ij} | 0 < \rho_G \le 1\} \qquad \rho_G^- = \{g_{ij} | -1 \le \rho_G < 0\} \qquad \rho_G^0 = \{g_{ij} | \rho_G = 0\}$$

Definition 3—Similarity relation: According to the specific universe of discourse classification, a similarity relation of the decision attributes $d \in D$ is denoted as U/D

$$S(D) = U|D = \{[x_i]_D | x_i \in U, V_{d_k} > V_{d_l}\}$$

Example:

$$S(d_1) = U/d_1 = \{\{x_1\},\{x_4\},\{x_2 x_3, x_5\}\}$$
$$S(d_2) = U/d_2 = \{\{x_3\},\{x_5\},\{x_2\},\{x_1, x_4\}\}$$

Definition 4—Potential relation between general attribute and decision attributes: The decision attributes in the information system are an ordered set, therefore, the attribute values will have an ordinal relation defined as follows:

$$\sigma_{GD} = Cov(g_i, d_k) \qquad \rho_{GD} = \frac{\sigma_{GD}}{\sqrt{Var(g_i)}\sqrt{Var(d_k)}}$$

Then:

$$F(G,D) = \begin{cases} \rho_{GD}^{+} : 0 < \rho_{GD} \le 1 \\ \rho_{GD}^{-} : -1 \le \rho_{GD} < 0 \\ \rho_{GD}^{0} : \rho_{GD} = 0 \end{cases}$$

Second: Generated rough associational rule—Definition 1: The first step in this study, we have found the potential relation between general attribute and decision attributes, hence in the step, the object is to generated rough associational rule. To consider other attributes and the core attribute of ordinal-scale data as the highest decision-making attributes is hereby to establish the decision table and the ease to generate rules, as shown in Table 5. $DT = (U, Q)$, where $U = \{x_1, x_2, \cdots x_n\}$ is a finite set of objects, Q is usually divides into two parts, $G = \{g_1, g_2, \cdots g_m\}$ is a finite set of general attributes/criteria, $D = \{d_1, d_2, \cdots d_l\}$ is a set of decision attributes. $f_g = U \times G \to V_g$ is called the information function, V_g is the domain of the attribute/criterion g, and f_g is a total function such that $f(x, g) \in V_g$ for each $g \in Q$; $x \in U$. $f_d = U \times D \to V_d$ is called the sorting decision-making information function, V_d is the domain of the decision attributes/criterion d, and f_d is a total function such that $f(x, d) \in V_d$ for each $d \in Q$; $x \in U$.

Then:

$$f_{g_1} = \{\text{Price, Brand}\} \qquad f_{g_2} = \{\text{Seen on shelves, Advertising}\}$$
$$f_{g_3} = \{\text{purchase by promotions, will not purchase by promotions}\}$$
$$f_{g_4} = \{\text{Convenience Stores, Hypermarkets}\}$$

Definition 2: According to the specific universe of discourse classification, a similarity relation of the general attributes is denoted by U/G. All of the similarity relation is denoted by $K = (U, R_1, R_2 \cdots R_{m-1})$.

$$U|G = \{[x_i]_G | x_i \in U\}$$

Example:

$$R_1 = \frac{U}{g_1} = \{\{x_1, x_2, x_5\}, \{x_3, x_4\}\} \qquad R_6 = \frac{U}{g_2 g_4} = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\}$$

$$\vdots \qquad\qquad \vdots$$

$$R_5 = \frac{U}{g_1 g_3} = \{\{x_1, x_2, x_5\}, \{x_3, x_4\}\} \qquad R_{m-1} = \frac{U}{G} = \{\{x_1\}, \{x_2, x_5\}, \{x_3, x_4\}\}$$

Table 5. Decision-making

Q	General attributes				Decision attributes	
U	Product Features g_1	Product Information Source g_2	Consumer Behavior g_3	Channels g_4	Rank	Brand
x_1	Price	Seen on shelves	purchase by promotions	Convenience Stores	4	d_1
x_2	Price	Advertising	purchase by promotions	Hypermarkets	1	d_1
x_3	Brand	Seen on shelves	will not purchase by promotions	Convenience Stores	1	d_1
x_4	Brand	Seen on shelves	will not purchase by promotions	Convenience Stores	3	d_1
x_5	Price	Advertising	purchase by promotions	Hypermarkets	1	d_1

Definition 3: According to the similarity relation, and then finding the reduct and core. If the attribute g which were ignored from G, the set G will not be affected; thereby, g is an unnecessary attribute, we can reduct it. $R \subseteq G$ and $\forall_g \in R$. A similarity relation of the general attributes from the decision table is denoted by $ind(G)$. If $ind(G) = ind(G - g_1)$, then g_1 is the reduct attribute, and if $ind(G) \neq ind(G - g_1)$, then g_1 is the core attribute.

Example:

$$U|ind(G) = \{\{x_1\}, \{x_2, x_5\}, \{x_3, x_4\}\}$$

$$U|ind(G - g_1) = U|(\{g_2, g_3, g_4\}) = \{\{x_1\}, \{x_2, x_5\}, \{x_3, x_4\}\} = U|ind(G)$$

$$U|ind(G - g_1 g_3) = U|(\{g_2, g_4\}) = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\} \neq U|ind(G)$$

When g_1 is considered alone, g_1 is the reduct attribute, but when g_1 and g_3 are considered simultaneously, g_1 and g_3 are the core attributes.

Definition 4: The lower approximation, denoted as $\underline{G}(X)$, is defined as the union of all these elementary sets, which are contained in $[x_i]_G$. More formally,

$$\underline{G}(X) = \cup\left\{[x_i]_G \in \frac{U}{G} \middle| [x_i]_G \subseteq X\right\}$$

The upper approximation, denoted as $\overline{G}(X)$, is the union of these elementary sets, which have a non-empty intersection with $[x_i]_G$. More formally:

$$\overline{G}(X) = \cup\left\{[x_i]_G \subseteq \frac{U}{G} \middle| [x_i]_G \cap X \neq \phi\right\}$$

The difference $Bn_G(X) = \overline{G}(X) - \underline{G}(X)$ is called the boundary of $[x_i]_G$.

Example: $\{x_1, x_2, x_4\}$ are those customers that we are interested in, thereby $\underline{G}(X) = \{x_1\}$, $\overline{G}(X) = \{x_1, x_2, x_3, x_4, x_5\}$ and $Bn_G(X) = \{x_2, x_3, x_4, x_5\}$.

Definition 5: Rough set-based association rules.

$$\frac{\{x_1\}}{g_1 g_3} : g_{1_1} \cap g_{3_1} \Rightarrow d^1_{d_1} = 4 \qquad \frac{\{x_1\}}{g_1 g_2 g_3 g_4} : g_{1_1} \cap g_{2_1} \cap g_{3_1} \cap g_{4_1} \Rightarrow d^1_{d_1} = 4$$

Algorithm-Step1

```
Input:
Information System (IS);
Output:
{Potential relation};
Method:
 1. Begin
 2.   IS = (U, Q);
 3.    x_1, x_2, ..., x_n ∈ U ; /* where x_1, x_2, ..., x_n are the objects of
       set U */
 4.    G, D ⊂ Q ; /* Q is divided into two parts G and D */
 5.      g_1, g_2, ..., g_i ∈ G ; /* where g_1, g_2, ..., g_i are the elements
         of set G */
 6.      d_1, d_2, ..., d_k ∈ D ; /* where d_1, d_2, ..., d_k are the elements
         of set D */
 7.   For each g_i and d_k do;
 8.     compute f(x, g) and f(x, d); /* compute the information
         function in IS as described in definition1*/
 9.     compute σ_G ; /* compute the quantity attribute
         covariance in IS as described in definition2*/
```

```
10.    compute ρ_G ; /* compute the quantity attribute
         correlation coefficient in IS as described in
         definition2*/
11.    compute S(D) and S(D) ; /* compute the similarity
         relation in IS as described in definition3*/
12.    compute F(G,D) ; /* compute the potential relation
         as described in definition4*/
13.  Endfor;
14.    Output {Potential relation};
15.End;
```

Algorithm-Step2

```
Input:
Decision Table (DT);
Output:
{Classification Rules};
Method:
 1. Begin
 2.   DT = (U,Q) ;
 3.     x_1,x_2,⋯x_n ∈ U ; /* where x_1,x_2,⋯x_n are the objects of
        set U */
 4.     Q = (G,D) ;
 5.      g_1,g_2,⋯,g_m ∈ G ; /* where g_1,g_2,⋯,g_m are the
          elements of set G */
 6.      d_1,d_2,⋯,d_l ∈ D ; /* where d_1,d_2,⋯,d_l are the "trust
          value" generated in Step1*/
 7.  For each d_l do;
 8.    compute f(x,g) ; /* compute the information function
        in DT as described in definition1*/
 9.    compute R_m ; /* compute the similarity relation in
        DT as described in definition2*/
10.    compute ind(G) ; /* compute the relative reduct of
        DT as described in definition3*/
11.    compute ind(G − g_m) ; /* compute the relative reduct
        of the elements for element m as described in
        definition3*/
12.    compute G(X) ; /* compute the lower-approximation
        of DT as described in definition4*/
13.    compute Ḡ(X) ; /* compute the upper-approximation
        of DT as described in definition4*/
14.    compute Bn_G(X) ; /* compute the bound of DT as
        described in definition4*/
15.  Endfor;
16.    Output {Association Rules};
17.End;
```

4 Conclusion and Future Works

The quantitative data are popular in practical databases; a natural extension is finding association rules from quantitative data. To solve this problem, previous research partitioned the value of a quantitative attribute into a set of intervals so that the traditional algorithms for nominal data could be applied [1]. In addition, most of the techniques used for finding association rule scan the whole data set, evaluate all possible rules, and retain only the rules that have support and confidence greater than thresholds [3]. The new association rule algorithm, which tries to combine with rough set theory to provide more easily explained rules for the user. In the research, we use a two-step algorithm to find the relative association rules. It will be easier for the user to find the association. Because, in the first step, we find out the relationship between the two quantities attribute data, and then we find whether the ordinal scale data has a potential relationship with those quantities attribute data. It can avoid human error caused by lack of experience in the process that quantities attribute data transform to categorical data. At the same time, we known the potential relationship between the quantities attribute data and ordinal-scale data. In the second step, we use the rough set theory benefit, which has the ability to handle uncertainty in the classing process, and find out the relative association rules. The user in mining association rules does not have to set a threshold and generate all association rules that have support and confidence greater than the user-specified thresholds. In this way, the association rules will be a relative association rules. The new association rule algorithm, which tries to combine with the rough set theory to provide more easily explained rules for the user. For the convenience of the users, to design an expert support system will help to improve the efficiency of the user.

Acknowledgements. This research was funded by the National Science Council, Taiwan, Republic of China, under contract NSC 100-2410-H-032 -018-MY3.

References

1. Chen, Y.L., Weng, C.H.: Mining association rules from imprecise ordinal data. Fuzzy Sets and Systems 159, 460–474 (2008)
2. Lian, W., Cheung, D.W., Yiu, S.M.: An efficient algorithm for finding dense regions for mining quantitative association rules. Computers and Mathematics with Applications 50(3-4), 471–490 (2005)
3. Liao, S.H., Chen, Y.J.: A rough association rule is applicable for knowledge discovery. In: IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2009), ShangHai, China (2009)
4. Liu, G., Zhu, Y.: Credit Assessment of Contractors: A Rough Set Method. Tsinghua Science & Technology 11, 357–363 (2006)
5. Pawlak, Z.: Rough sets, decision algorithms and Bayes' theorem. European Journal of Operational Research 136, 181–189 (2002)
6. Rebolledo, M.: Rough intervals—enhancing intervals for qualitative modeling of technical systems. Artificial Intelligence 170(8-9), 667–668 (2006)

Scalable Data Clustering: A Sammon's Projection Based Technique for Merging GSOMs

Hiran Ganegedara and Damminda Alahakoon

Cognitive and Connectionist Systems Laboratory,
Faculty of Information Technology,
Monash University, Australia 3800
{hiran.ganegedara,damminda.alahakoon}@monash.edu
http://infotech.monash.edu/research/groups/ccsl/

Abstract. Self-Organizing Map (SOM) and Growing Self-Organizing Map (GSOM) are widely used techniques for exploratory data analysis. The key desirable features of these techniques are applicability to real world data sets and the ability to visualize high dimensional data in low dimensional output space. One of the core problems of using SOM/GSOM based techniques on large datasets is the high processing time requirement. A possible solution is the generation of multiple maps for subsets of data where the subsets consist of the entire dataset. However the advantage of topographic organization of a single map is lost in the above process. This paper proposes a new technique where Sammon's projection is used to merge an array of GSOMs generated on subsets of a large dataset. We demonstrate that the accuracy of clustering is preserved after the merging process. This technique utilizes the advantages of parallel computing resources.

Keywords: Sammon's projection, growing self organizing map, scalable data mining, parallel computing.

1 Introduction

Exploratory data analysis is used to extract meaningful relationships in data when there is very less or no priori knowledge about its semantics. As the volume of data increases, analysis becomes increasingly difficult due to the high computational power requirement. In this paper we propose an algorithm for exploratory data analysis of high volume datasets.

The Self-Organizing Map (SOM)[12] is an unsupervised learning technique to visualize high dimensional data in a low dimensional output spacel. SOM has been successfully used in a number of exploratory data analysis applications including high volume data such as climate data analysis[11], text clustering[16] and gene expression data[18]. The key issue with increasing data volume is the high computational time requirement since the time complexity of the SOM is in the order of $O(n^2)$ in terms of the number of input vectors n[16]. Another challenge is the determination of the shape and size of the map. Due to the high

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 193–202, 2011.
© Springer-Verlag Berlin Heidelberg 2011

volume of the input, identifying suitable map size by trial and error may become impractical.

A number of algorithms have been developed to improve the performance of SOM on large datasets. The Growing Self-Organizing Map (GSOM)[2] is an extension to the SOM algorithm where the map is trained by starting with only four nodes and new nodes are grown to accommodate the dataset as required. The degree of spread of the map can be controlled by the parameter *spread factor*. GSOM is particularly useful for exploratory data analysis due to its ability to adapt to the structure of data so that the size and the shape of the map need not be determined in advance. Due to the initial small number of nodes and the ability to generate nodes only when required, the GSOM demonstrates faster performance over SOM[3]. Thus we considered GSOM more suited for exploratory data analysis.

Emergence of parallel computing platforms has the potential to provide the massive computing resources for large scale data analysis. Although several serial algorithms have been proposed for large scale data analysis using SOM[15] [8], such algorithms tend to perform less efficiently as the input data volume increases. Thus several parallel algorithms for SOM and GSOM have been proposed in [16] [13] and [20]. [16] and [13] are developed to operate on sparse datasets, with the principal application area being textual classification. In addition, [13] needs access to shared memory during the SOM training phase. Both [16] and [20] rely on an expensive initial clustering phase to distribute data to parallel computing nodes. In [20], a merging technique is not suggested for the maps generated in parallel.

In this paper, we develop a generic scalable GSOM data clustering algorithm which can be trained in parallel and merged using Sammon's projection[17]. Sammon's projection is a nonlinear mapping technique from high dimensional space to low dimensional space. GSOM training phase can be made parallel by partitioning the dataset and training a GSOM on each data partition. Sammon's projection is used to merge the separately generated maps. The algorithm can be scaled to work on several computing resources in parallel and therefore can utilize the processing power of parallel computing platforms. The resulting merged map is refined to remove redundant nodes that may occur due to the data partitioning method.

This paper is organized as follows. Section 2 describes SOM, GSOM and Sammon's Projection algorithms, the literature related to the work presented in this paper. Section 3 describes the proposed algorithm in detail and Section 4 describes the results and comparisons. The paper is concluded with Section 5, stating the implications of this work and possible future enhancements.

2 Background

2.1 Self-Organizing Map

The SOM is an unsupervised learning technique which maps high dimensional input space to a low dimensional output lattice. Nodes are arranged in the

low dimensional lattice such that the distance relationships in high dimensional space are preserved. This topology preservation property can be used to identify similar records and to cluster the input data. Euclidean distance is commonly used for distance calculation using

$$d_{ij} = |x_i - x_j| \ . \tag{1}$$

where d_{ij} is the distance between vectors x_i and x_j. For each input vector, the Best Matching Unit (BMU) x_k is found using Eq. (5) such that d_{ik} is minimum when x_i is the input vector and k is any node in the map. Neighborhood weight vectors of the BMU are adjusted towards the input vector using

$$w_k^* = w_k + \alpha h_{ck}[x_i - w_k] \ . \tag{2}$$

where w_k^* is the new weight vector of node k, w_k is the current weight, α is the learning rate, h_{ck} is the neighborhood function and x_i is the input vector. This process is repeated for a number of iterations.

2.2 Growing Self-Organizing Map

A key decision in SOM is the determination of the size and the shape of the map. In order to determine these parameters, some knowledge about the structure of the input is required. Otherwise trial and error based parameter selection can be applied. SOM parameter determination could become a challenge in exploratory data analysis since structure and nature of input data may not be known. The GSOM algorithm is an extension to SOM which addresses this limitation. The GSOM starts with four nodes and has two phases, a growing phase and a smoothing phase.

In the growing phase, each input vector is presented to the network for a number of iterations. During this process, each node accumulates an error value determined by the distance between the BMU and the input vector. When the accumulated error is greater than the growth threshold, nodes are grown if the BMU is a boundary node. The growth threshold GT is determined by the spread factor SF and the number of dimensions D. GT is calculated using

$$GT = -D \times \ln SF \ . \tag{3}$$

For every input vector, the BMU is found and the neighborhood is adapted using Eq. (2).

The smoothing phase is similar to the growing phase, except for the absence of node growth. This phase distributes the weights from the boundary nodes of the map to reduce the concentration of hit nodes along the boundary.

2.3 Sammon's Projection

Sammon's projection is a nonlinear mapping algorithm from high dimensional space onto a low dimensional space such that topology of data is preserved. The

Sammon's projection algorithm attempts to minimize Sammon's stress E over a number of iterations given by

$$E = \frac{1}{\sum_{\mu=1}^{n-1} \sum_{v=\mu+1}^{n} d*(\mu, v)} \times \sum_{\mu=1}^{n-1} \sum_{v=\mu+1}^{n} \frac{[d*(\mu, v) - d(\mu, v)]^2}{d*(\mu, v)} . \quad (4)$$

Sammon's projection cannot be used on high volume input datasets due to its time complexity being $O(n^2)$. Therefore as the number of input vectors, n increases, the computational requirement grows exponentially. This limitation has been addressed by integrating Sammon's projection with neural networks[14].

3 The Parallel GSOM Algorithm

In this paper we propose an algorithm which can be scaled to suit the number of parallel computing resources. The computational load on the GSOM primarily depends on the size of the input dataset, the number of dimensions and the spread factor. However the number of dimensions is fixed and the spread factor depends on the required granularity of the resulting map. Therefore the only parameter that can be controlled is the size of the input, which is the most significant contributor to time complexity of the GSOM algorithm.

The algorithm consists of four stages, data partitioning, parallel GSOM training, merging and refining. Fig. (3) shows the high level view of the algorithm.

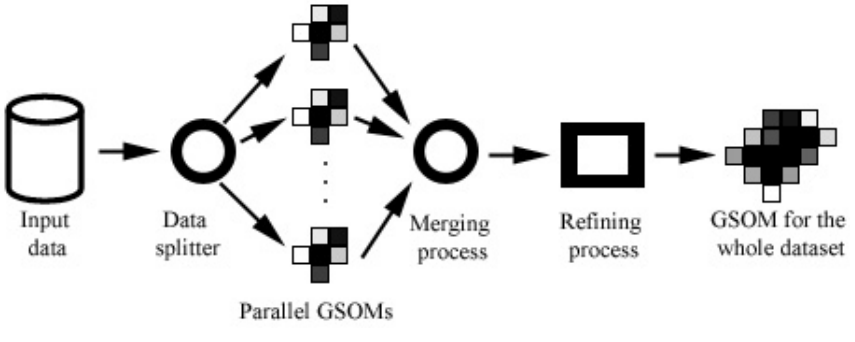


Fig. 1. The Algorithm

3.1 Data Partitioning

The input dataset has to be partitioned according to the number of parallel computing resources available. Two possible partitioning techniques are considered in the paper. First is random partitioning where the dataset is partitioned randomly without considering any property in the dataset. Random splitting could be used if the dataset needs to be distributed evenly across the GSOMs. Random partitioning has the advantage of lower computational load although even spread is not always guaranteed.

The second technique is splitting based on very high level clustering [19] [20]. Using this technique, possible clusters in data can be identified and SOMs or GSOMs are trained on each cluster. These techniques help in decreasing the number of redundant neurons in the merged map. However the initial clustering process requires considerable computational time for very large datasets.

3.2 Parallel GSOM Training

After the data partitioning process, a GSOM is trained on each partition in a parallel computing environment. The spread factor and the number of growing phase and smoothing phase iterations should be consistent across all the GSOMs. If random splitting is used, partitions could be of equal size if each computing unit in the parallel environment has the same processing power.

3.3 Merging Process

Once the training phase is complete, output GSOMs are merged to create a single map representing the entire dataset. Sammon's projection is used as the merging technique due to the following reasons.

a. Sammon's projection does not include learning. Therefore the merged map will preserve the accumulated knowledge in the neurons of the already trained maps. In contrast, using SOM or GSOM to merge would result in a map that is biased towards clustering of the separate maps instead of the input dataset.
b. Sammon's projection will better preserve topology of the map compared to GSOM as shown in results.
c. Due to absence of learning, Sammon's projection performs faster than techniques with learning.

Neurons generated in maps resulting from the GSOMs trained in parallel are used as input for the Sammon's projection algorithm which is run over a number of iterations to organize the neurons in topological order. This enables the representation of the entire input dataset in the merged map with topology preserved.

3.4 Refining Process

After merging, the resulting map is refined to remove any redundant neurons. In the refining process, nearest neighbor based distance measure is used to merge any redundant neurons. The refining algorithm is similar to [6] where, for each node in the merged map, the distance between the nearest neighbor coming from the same source map, d_1, and the distance between the nearest neighbor from the other maps, d_2, as described Eq. (5). Neurons are merged if

$$\frac{d_1}{d_2} \geq \beta e^{SF} \tag{5}$$

where β is the scaling factor and SF is the spread factor used for the GSOMs.

4 Results

We used the proposed algorithm on several datasets and compared the results with a single GSOM trained on the same datasets as a whole. A multi core computer was used as the parallel computing environment where each core is considered a computing node. Topology of the input data is better preserved in Sammon's projection than GSOM. Therefore in order to compensate for the effect of Sammon's projection, the map generated by the GSOM trained on the whole dataset was projected using Sammon's projection and included in the comparison.

4.1 Accuracy

Accuracy of the proposed algorithm was evaluated using breast cancer Wisconsin dataset from UCI Machine Learning Repository[9]. Although this dataset may not be considered as large, it provides a good basis for cluster evaluation[5]. The dataset has 699 records each having 9 numeric attributes and 16 records with missing attribute values were removed. The parallel run was done on two computing nodes. Records in the dataset are classified as 65.5% benign and 34.5% malignant.

The dataset was randomly partitioned to two segments containing 341 and 342 records. Two GSOMs were trained in parallel using the proposed algorithm and another GSOM was trained on the whole dataset. All the GSOM algorithms were trained using a spread factor of 0.1, 50 growing iterations and 100 smoothing iterations. Results were evaluated using three measures for accuracy, DB index, cross cluster analysis and topology preservation.

DB Index. DB Index[1] was used to evaluate the clustering of the map for different numbers of clusters. K-means[10] algorithm was used to cluster the map for k values from 2 to $\sqrt{n}$, n being the number of nodes in the map. For exploratory data analysis, DB Index is calculated for each k and the value of k for which DB Index is minimum, is the optimum number of clusters.

Table 1 shows that the DB Index values are similar for different k values across the three maps. It indicates similar weight distributions across the maps.

Table 1. DB index comparison

k	GSOM	GSOM with Sammon's Projection	Parallel GSOM
2	0.400	0.285	0.279
3	0.448	0.495	0.530
4	0.422	0.374	0.404
5	0.532	0.381	0.450
6	0.545	0.336	0.366

Cross Cluster Analysis. Cross cluster analysis was performed between two sets of maps. Table 2 shows how the input vectors are mapped to clusters of GSOM and the parallel GSOM. It can be seen that 97.49% of the data items mapped to cluster 1 of the GSOM are mapped to cluster 1 of the parallel GSOM, similarly 90.64% of the data items in cluster 2 of the GSOM are mapped to the corresponding cluster in the parallel GSOM.

Table 2. Cross cluster comparison of parallel GSOM and GSOM

		Parallel GSOM	
		Cluster 1	Cluster 2
GSOM	Cluster 1	97.49%	2.51%
	Cluster 2	9.36%	90.64%

Table 3 shows the comparison between GSOM with Sammon's projection and the parallel GSOM. Due to better topology preservation, the results are slightly better for the proposed algorithm.

Table 3. Cross cluster comparison of parallel GSOM and GSOM with Sammon's projection

		Parallel GSOM	
		Cluster 1	Cluster 2
GSOM with Sammon's Projection	Cluster 1	98.09%	1.91%
	Cluster 2	8.1%	91.9%

Topology Preservation. A comparison of the degree of topology preservation of the three maps are shown in Table 4. Topographic product [4] is used as the measure of topology preservation. It is evident that maps generated using Sammon's projection have better topology preservation leading to better results in terms of accuracy. However the topographic product scales nonlinearly with the number of neurons. Although it may lead to inconsistencies, the topographic product provides a reasonable measure to compare topology preservation in the maps.

Table 4. Topographic product

GSOM	GSOM with Sammon's Projection	Parallel GSOM
-0.01529	0.00050	0.00022

Similar results were obtained for other datasets, for which results are not shown due to space constraint. Fig. 2 shows clustering of GSOM, GSOM with Sammon's projection and the parallel GSOM. It is clear that the map generated by the proposed algorithm is similar in topology to the GSOM and the GSOM with Sammon's projection.

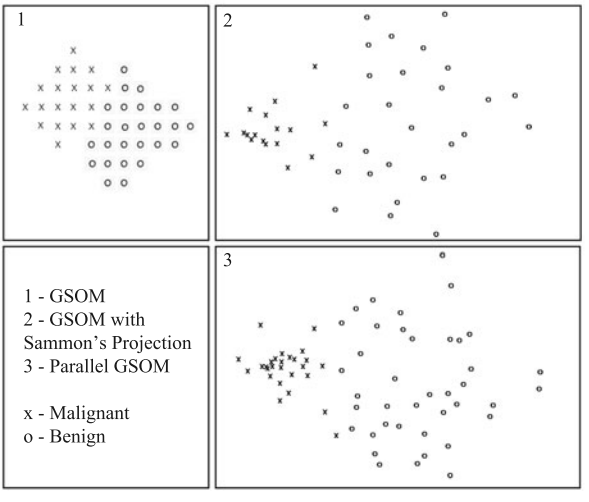


Fig. 2. Clustering of maps for breast cancer dataset

4.2 Performance

The key advantage of a parallel algorithm over a serial algorithm is better performance. We used a dual core computer as a the parallel computing environment where two threads can simultaneously execute in the two cores. The execution time decreases exponentially with the number of computing nodes available.

Execution time of the algorithm was compared using three datasets, breast cancer dataset used for accuracy analysis, the mushroom dataset from [9] and muscle regeneration dataset (9GDS234) from [7]. The mushroom dataset has 8124 records and 22 categorical attributes which resulted in 123 attributes when converted to binary. The muscle regeneration dataset contains 12488 records with 54 attributes. The mushroom and muscle regeneration datasets provided a better view of the algorithms performance for large datasets. Table 5 summarizes

Table 5. Execution Time

	Breast cancer	Mushroom	Micro-array
GSOM	4.69	1141	1824
Parallel GSOM	2.89	328	424

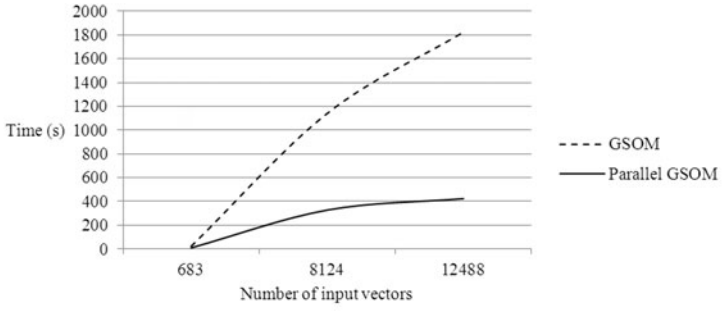


Fig. 3. Execution time graph

the results for performance n terms of execution time. Fig. 3 shows the results in a graph.

5 Discussion

We propose a scalable algorithm for exploratory data analysis using GSOM. The proposed algorithm can make use of the high computing power provided by parallel computing technologies. This algorithm can be used on any real-life dataset without any knowledge about the structure of the data. When using SOM to cluster large datasets, two parameters should be specified, width and hight of the map. User specified width and height may or may not suite the dataset for optimum clustering. This is especially the case with the proposed technique due to the user having to specify suitable SOM size and shape for selected data subsets. In the case for large scale datasets, using a trial and error based width and hight selection may not be possible. GSOM has the ability to grow the map according to the structure of the data. Since the same *spread factor* is used across all subsets, comparable GSOMs will be self generated with data driven size and shape. As a result, although it it possible to use this technique on SOM, it is more appropriate for GSOM. It can be seen that the proposed algorithm is several times efficient than the GSOM and gives the similar results in terms of accuracy. The efficiency of the algorithm grows exponentially with the number of parallel computing nodes available. As a future development, the refining method will be fine tuned and the algorithm will be tested on a distributed grid computing environment.

References

1. Ahmad, N., Alahakoon, D., Chau, R.: Cluster identification and separation in the growing self-organizing map: application in protein sequence classification. Neural Computing & Applications 19(4), 531–542 (2010)
2. Alahakoon, D., Halgamuge, S., Srinivasan, B.: Dynamic self-organizing maps with controlled growth for knowledge discovery. IEEE Transactions on Neural Networks 11(3), 601–614 (2000)

3. Amarasiri, R., Alahakoon, D., Smith-Miles, K.: Clustering massive high dimensional data with dynamic feature maps, pp. 814–823. Springer, Heidelberg
4. Bauer, H., Pawelzik, K.: Quantifying the neighborhood preservation of self-organizing feature maps. IEEE Transactions on Neural Networks 3(4), 570–579 (1992)
5. Bennett, K., Mangasarian, O.: Robust linear programming discrimination of two linearly inseparable sets. Optimization methods and software 1(1), 23–34 (1992)
6. Chang, C.: Finding prototypes for nearest neighbor classifiers. IEEE Transactions on Computers 100(11), 1179–1184 (1974)
7. Edgar, R., Domrachev, M., Lash, A.: Gene expression omnibus: Ncbi gene expression and hybridization array data repository. Nucleic acids research 30(1), 207 (2002)
8. Feng, Z., Bao, J., Shen, J.: Dynamic and adaptive self organizing maps applied to high dimensional large scale text clustering, pp. 348–351. IEEE (2010)
9. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
10. Hartigan, J.: Clustering algorithms. John Wiley & Sons, Inc. (1975)
11. Hewitson, B., Crane, R.: Self-organizing maps: applications to synoptic climatology. Climate Research 22(1), 13–26 (2002)
12. Kohonen, T.: The self-organizing map. Proceedings of the IEEE 78(9), 1464–1480 (1990)
13. Lawrence, R., Almasi, G., Rushmeier, H.: A scalable parallel algorithm for self-organizing maps with applications to sparse data mining problems. Data Mining and Knowledge Discovery 3(2), 171–195 (1999)
14. Lerner, B., Guterman, H., Aladjem, M., Dinsteint, I., Romem, Y.: On pattern classification with sammon's nonlinear mapping an experimental study* 1. Pattern Recognition 31(4), 371–381 (1998)
15. Ontrup, J., Ritter, H.: Large-scale data exploration with the hierarchically growing hyperbolic som. Neural networks 19(6-7), 751–761 (2006)
16. Roussinov, D., Chen, H.: A scalable self-organizing map algorithm for textual classification: A neural network approach to thesaurus generation. Communication Cognition and Artificial Intelligence 15(1-2), 81–111 (1998)
17. Sammon Jr., J.: A nonlinear mapping for data structure analysis. IEEE Transactions on Computers 100(5), 401–409 (1969)
18. Sherlock, G.: Analysis of large-scale gene expression data. Current Opinion in Immunology 12(2), 201–205 (2000)
19. Yang, M., Ahuja, N.: A data partition method for parallel self-organizing map, vol. 3, pp. 1929–1933. IEEE
20. Zhai, Y., Hsu, A., Halgamuge, S.: Scalable dynamic self-organising maps for mining massive textual data, pp. 260–267. Springer, Heidelberg

A Generalized Subspace Projection Approach for Sparse Representation Classification

Bingxin Xu and Ping Guo

Image Processing and Pattern Recognition Laboratory
Beijing Normal University, Beijing 100875, China
xbing@mail.bnu.edu.cn, pguo@ieee.org

Abstract. In this paper, we propose a subspace projection approach for sparse representation classification (SRC), which is based on Principal Component Analysis (PCA) and Maximal Linearly Independent Set (MLIS). In the projected subspace, each new vector of this space can be represented by a linear combination of MLIS. Substantial experiments on Scene15 and CalTech101 image datasets have been conducted to investigate the performance of proposed approach in multi-class image classification. The statistical results show that using proposed subspace projection approach in SRC can reach higher efficiency and accuracy.

Keywords: Sparse representation classification, subspace projection, multi-class image classification.

1 Introduction

Sparse representation has been proved an extremely powerful tool for acquiring, representing, and compressing high-dimensional signals [1]. Moreover, the theory of compressive sensing proves that sparse or compressible signals can be accurately reconstructed from a small set of incoherent projections by solving a convex optimization problem [6]. While these successes in classical signal processing application are inspiring, in computer vision we are often more interested in the content or semantics of an image rather than a compact, high-fidelity representation [1]. In literatures, sparse representation has been applied to many computer vision tasks, including face recognition [2], image super-resolution [3], data clustering [4] and image annotation [5]. In the application of sparse representation in computer vision, sparse representation classification framework [2] is a novel idea which cast the recognition problem as one of classifying among multiple linear regression models and applied in face recognition successfully.

However, to successfully apply sparse representation to computer vision tasks, an important problem is how to correctly choose the basis for representing the data. While in the previous research, there is little study of this problem. In reference [2], the authors just emphasize the training samples must be sufficient and there is no specific instruction for how to choose them can achieve well results. They only use the entire training samples of face images and the number of training samples is decided by different image datasets. In this paper, we try

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 203–210, 2011.
© Springer-Verlag Berlin Heidelberg 2011

to solve this problem by proposing a subspace projection approach, which can guide the selection of training data for each class and explain the rationality of sparse representation classification in vector space.

The ability of sparse representation to uncover semantic information derives in part from a simple but important property of the data. That is although the images or their features are naturally very high dimensional , in many applications images belonging to the same class exhibit degenerate structure which means they lie on or near low dimensional subspaces [1]. The proposed approach in this paper is based on this property of data and applied in multi-class image classification. The motivation is to find a collection of representative samples in each class's subspace which is embedded in the original high dimensional feature space. The main contribution of this paper can be summarized as follows:

1. Using a simple linear method to search the subspace of each class data is proposed, the original feature space is divided into several subspaces and each category belongs to a subspace.
2. A basis construction method by applying the theory of Maximal Linearly Independent Set is proposed. Based on linear algebra knowledge, for a fixed vector space, only a portion of vectors are sufficient to represent any others which belong to the same space.
3. Experiments are conducted for multi-class image classification with two standard bench marks, which are Scene15 and CalTech101 datasets. The performance of proposed method (subspace projection sparse representation classification, SP_SRC) is compared with sparse representation classification (SRC), nearest neighbor (NN) and support vector machine (SVM).

2 Sparse Representation Classification

Sparse representation classification assumes that training samples from a single class do lie on a subspace [2]. Therefore, any test sample from one class can be represented by a linear combination of training samples in the same class. If we arrange the whole training data from all the classes in a matrix, the test data can be seen as a sparse linear combination of all the training samples.

Specifically, given N_i training samples from the i-th class, the samples are stacked as columns of a matrix $\mathbf{F}_i = [f_{i,1}, f_{i,2}, \ldots, f_{i,N_i}] \in R^{m \times N_i}$. Any new test sample $\mathbf{y} \in R^m$ from the same class will approximately lie in the linear subspace of the training samples associated with class i [2]:

$$\mathbf{y} = x_{i,1} f_{i,1} + x_{i,2} f_{i,2} + \ldots + x_{i,N_i} f_{i,N_i}, \tag{1}$$

where $x_{i,j}$ is the coefficient of linear combination, $j = 1, 2, ..., N_i$. $\mathbf{y}$ is the test sample's feature vector which is extracted by the same method with training samples.

Since the class i of the test sample is unknown, a new matrix $\mathbf{F}$ is defined by concatenation the $N = \sum_{i=1}^{c} N_i$ training samples of all c classes:

$$\mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_c] = [f_{1,1}, f_{1,2}, ..., f_{c,N_c}]. \tag{2}$$

Then the linear representation of $\mathbf{y}$ can be rewritten in terms of all the training samples as

$$\mathbf{y} = \mathbf{F}\mathbf{x} \in R^m, \tag{3}$$

where $\mathbf{x} = [0, ..., 0, x_{i,1}, x_{i,2}, ..., x_{i,N_i}, 0, ...0]^T \in R^N$ is the coefficient vector whose entries are zero except those associated with i-th class. In the practical application, the dimension m of feature vector is far less than the number of training samples N. Therefore, equation (3) is an underdetermined equation. However, the additional assumption of sparsity makes solve this problem possible and practical [6]. A classical approach of solving $\mathbf{x}$ consists in solving the ℓ_0 norm minimization problem:

$$\min \|\mathbf{y}\text{-}\mathbf{F}\mathbf{x}\|_2 + \lambda \|x\|_0, \tag{4}$$

where λ is the regularization parameter and ℓ_0 norm counts the number of non-zero entries in $\mathbf{x}$ [7]. However, the above approach is not reasonable in practice because it is a NP-hard problem [8]. Fortunately, the theory of compressive sensing proves that ℓ_1-minimization can instead of the ℓ_0 norm minimization in solving the above problem. Therefore, equation (4) can be rewritten as:

$$\min \|\mathbf{y}\text{-}\mathbf{F}\mathbf{x}\|_2 + \lambda \|x\|_1, \tag{5}$$

This is a convex optimization problem which can be solved via classical approaches such as basis pursuit [7]. After computing the coefficient vector $\mathbf{x}$, the identity of $\mathbf{y}$ is defined:

$$\min r_i(y) = \|\mathbf{y} - \mathbf{F}_i \delta_i(x)\|_2, \tag{6}$$

where $\delta_i(x)$ is the part coefficients of $\mathbf{x}$ which associated with the i-th class.

3 Subspace Projection for Sparse Representation Classification

In the sparse representation classification (SRC) method, the key problem is whether and why the training samples are appropriate to represent the test data linearly. In reference [2], the authors said that given sufficient training samples of the i-th object class, any new test sample can be as a linear combination of the entire training data in this class. However, is that the more the better? Undoubtedly, through the increase of the training samples, the computation cost will also increase greatly. In the experiments of reference [2], the number of training data for each class is 7 and 32. These number of images are sufficient for face datasets but small for natural image classes due to the complexity of natural images. Actually, it is hard to estimate whether the number of training data of each class is sufficient quantitatively. What's more, in fixed vector space, the number of elements in maximal linearly independent set is also fixed. By adding more training samples will not influence the linear representation of test sample but increase the computing time. The proposed approach is trying to generate the appropriate training samples of each class for SRC.

3.1 Subspace of Each Class

For the application of SRC in multi-class image classification, feature vectors are extracted to represent the original images in feature space. For the entire image data, they are in a huge feature vector space which determined by the feature extraction method. In previous application methods, all the images are in the same feature space[17] [2]. However, different classes of images should lie on different subspaces which embedded in the original space. In the proposed approach, a simple linear principal component analysis (PCA) is used to find these subspaces for each class. PCA is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components [9]. In order to not destroy the linear relationship of each class, PCA is a better choice because it computes a linear transformation that maps data from a high dimensional space to a lower dimensional space. Specifically, $\mathbf{F}_i$ is an $m \times n_i$ matrix in the original feature space for i-th class where m is the dimension of feature vector and n_i is the number of training samples. After PCA processing, $\mathbf{F}_i$ is transformed into a $p \times n_i$ matrix $\mathbf{F}'_i$ which lie on the subspace of i-th class and p is the dimension of subspace.

3.2 Maximal Linearly Independent Set of Each Class

In the SRC, a test sample is assumed to be represented by a linear combination of the training samples in the same class. As mentioned in 3.1, after finding the subspace of each class, a vector subset is computed by MLIS in order to span the whole subspace. In linear algebra, a maximal linearly independent set is a set of linearly independent vectors that, in a linear combination, can represent every vector in a given vector space [10]. Given a maximal linearly independent set of vector space, every element of the vector space can be expressed uniquely as a finite linear combination of basis vectors. Specifically, in the subspace of $\mathbf{F}'_i$, if $p < n_i$, the number of elements in maximal linearly independent set is p [11]. Therefore, in the subspace of i-th class, only need p vectors to span the entire subspace. In proposed approach, the original training samples are substituted by the maximal linearly independent set. The retaining samples are redundant in the process of linear combination. The proposed multi-class image classification procedure is described as following Algorithm 1. The implementation of minimizing the ℓ_1 norm is based on the method in reference [12].

Algorithm 1: Image classification via subspace projection SRC (SP_SRC)

1. Input: feature space formed by training samples. $\mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_c] \in \mathbf{R}^{m \times N}$ for c classes and a test image feature vector $\mathbf{I}$.
2. For each $\mathbf{F}_i$, using PCA to form the subspace $\mathbf{F}'_i$ of i-th class.
3. For each subspace $\mathbf{F}'_i$, computing the maximal linearly independent set $\mathbf{F}''_i$. These subspaces form the new feature space $\mathbf{F} = [\mathbf{F}''_1, \mathbf{F}''_2, \ldots, \mathbf{F}''_c]$.
4. Computing $\mathbf{x}$ according to equation (5).
5. Output: identify the class number of test sample $\mathbf{I}$ with equation (6).

4 Experiments

In this section, experiments are conducted on publicly available datasets which are Scene15 [18] and CalTech101 [13] for image classification in order to evaluate the performance of proposed approach SP_SRC.

4.1 Parameters Setting

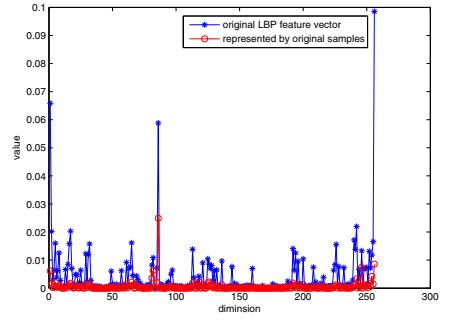
In the experiments, local binary pattern (LBP) [14] feature extraction method is used because of its effectiveness and ease of computation. The original LBP feature is used with dimension of 256. We compare our method with simple SRC and two classical algorithms, namely, nearest neighbor (NN) [15] and one-vs-one support vector machine (SVM) [16] which using the same feature vectors. In the proposed method, the most important two parameters are (i): the regularization parameter λ in equation (5). In the experiments, the performance is best when it is 0.1. (ii): the subspace dimension p. According to our observation, along with the increase of p, the performance is improved dramatically and then keep stable. Therefore, p is set to 30 in the experiments.

4.2 Experimental Results

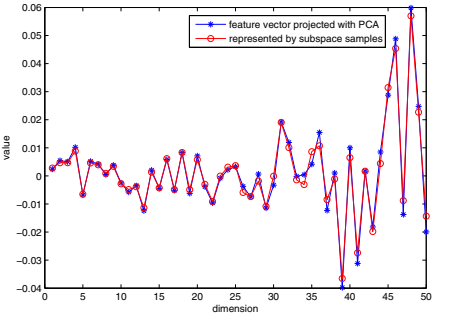
In order to illustrate the subspace projection approach proposed in this paper has better linear regression result, we compare the linear combination result between subspace projection SRC and original feature space SRC for a test sample. Figure 1(a) illustrates the linear representation result in the original LBP feature space. The blue line is the LBP feature vector for a test image and the red line is linear representation result by the training samples in the original LBP feature space. Figure 1(b) illustrates the linear representation result in projected subspace using the same method.

The classification experiments are conducted on two datasets to compare the performance of proposed method SP_SRC, SRC, NN and SVM classifier. To avoid contingency, each experiment is performed 10 times. At each time, we randomly selected a percentage of images from the datasets to be used as training samples. The remaining images are used for testing. The results presented represent the average of 10 times.

Scene15 Datasets. Scene15 contains totally 4485 images falling into 15 categories, with the number of images each category ranging from 200 to 400. The image content is diverse, containing not only indoor scene, such as bedroom, kitchen, but also outdoor scene, such as building and country. To compare with others' work, we randomly select 100 images per class as training data and use the rest as test data. The performance based on different methods is presented in Table 1. Moreover, the confusion matrix for scene is shown in Figure 2. From Table 1, we can find that in the LBP feature space, the SP_SRC has better results than the simple SRC, and outperforms other classical methods. Figure 2 shows the classification and misclassification status for each individual class. Our method performs outstanding for most classes.



(a)



(b)

Fig. 1. Regression results between different feature space. (a) linear regression in original feature space; (b) linear regression in the projected subspace.

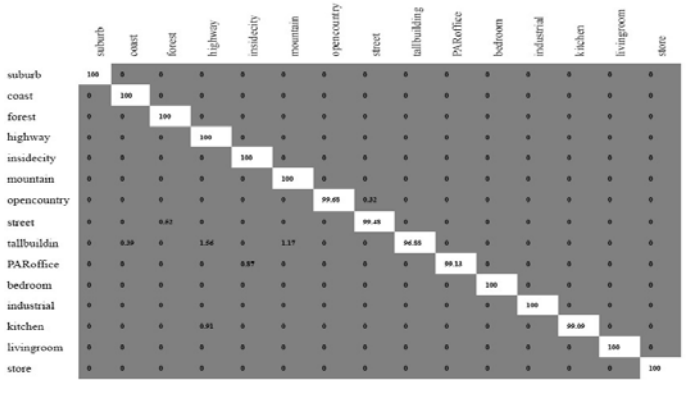


Fig. 2. Confusion Matrix on Scene15 datasets. In confusion matrix, the entry in the $i-th$ row and $j-th$ column is the percentage of images from class i that are misidentified as class j. Average classification rates for individual classes are presented along the diagonal.

Table 1. Precision rates of different classification method in Scene15 datasets

Classifier	SP_SRC	SRC	NN	SVM
Scene15	99.62%	55.96%	51.46%	71.64%

Table 2. Precision rates of different classification method in CalTech101 datasets

Classifier	SP_SRC	SRC	NN	SVM
CalTech101	99.74%	43.2%	27.65%	40.13%

CalTech101 Datasets. Another experiment is conducted on the popular caltech101 datasets, which consists of 101 classes. In this dataset, the numbers of images in different classes are varying greatly which range from several decades to hundreds. Therefore, in order to avoid data bias problem, a portion classes of dataset is selected which have similar number of samples. For demonstration the performance of SP_SRC, we select 30 categories from image datasets. The precision rates are represented in Table 2. From Table 2, we notice that our proposed method performs amazingly better than other methods for 30 categories. Comparing with Scene15 datasets, most methods' performance will decline for the increase of category number except the proposed method. This is due to that SP_SRC does not classify according to the inter-class differences, and it only depends on the intra-class representation degree.

5 Conclusion and Future Work

In this paper, a subspace projection approach is proposed which used in sparse representation classification framework. The proposed approach lays the theory foundation for the application of sparse representation classification. In the proposed method, each class samples are transformed into a subspace of the original feature space by PCA, and then computing the maximal linearly independent set of each subspace as basis to represent any other vector which in the same space. The basis of each class is just satisfied the precondition of sparse representation classification. The experimental results demonstrate that using the proposed subspace projection approach in SRC can achieve better classification precision rates than using all the training samples in original feature space. What is more, the computing time is also reduced because our method only use the maximal linearly independent set as basis instead of the entire training samples. It should be noted that the subspace of each class is different for different feature space. The relationship between a specified feature space and the subspaces of different classes still need to be investigated in the future. In addition, more accurate and fast computing way of ℓ_1-minimization is also a problem deserved to study.

Acknowledgment. The research work described in this paper was fully supported by the grants from the National Natural Science Foundation of China (Project No. 90820010, 60911130513). Prof. Ping Guo is the author to whom all correspondence should be addressed.

References

1. Wright, J., Ma, Y.: Sparse Representation for Computer Vision and Pattern Recognition. Proceedings of the IEEE 98(6), 1031–1044 (2009)
2. Wright, J., Yang, A.Y., Granesh, A.: Robust Face Recognition via Sparse Representation. IEEE Trans. on PAMI 31(2), 210–227 (2008)
3. Yang, J.C., Wright, J., Huang, T., Ma, Y.: Image superresolution as sparse representation of raw patches. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
4. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (2009)
5. Teng, L., Tao, M., Yan, S., Kweon, I., Chiwoo, L.: Contextual Decomposition of Multi-Label Image. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (2009)
6. Baraniuk, R.: Compressive sensing. IEEE Signal Processing Magazine 24(4), 118–124 (2007)
7. Candes, E.: Compressive sampling. In: Proceedings of the International Congress of Mathematicians, Madrid, Spain, pp. 1433–1452 (2006) AB2006
8. Donoho, D.: Compressed Sensing. IEEE Trans. on Information Theory 52(4), 1289–1306 (2006)
9. Jolliffe, I.T.: Principal Component Analysis, p. 487. Springer, Heidelberg (1986)
10. Blass, A.: Existence of bases implies the axiom of choice. Axiomatic set theory. Contemporary Mathematics 31, 31–33 (1984)
11. David, C.L.: Linear Algebra And It's Application, pp. 211–215 (2000)
12. Candes, E., Romberg, J.: ℓ_1-magic:Recovery of sparse signals via convex programming, http://www.acm.calltech.edu/l1magic/
13. Fei-fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (2004)
14. Ojala, T., Pietikainen, M.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE Trans.on PAMI 24(7), 971–987 (2002)
15. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. John Wiley and Sons (2001)
16. Hsu, C.W., Lin, C.J.: A Comparison of Methods for Multiclass Support Vector Machines. IEEE Trans. on Neural Networks 13(2), 415–425 (2002)
17. Yuan, Z., Bo, Z.: General Image Classifications based on sparse representaion. In: Proceedings of IEEE International Conference on Cognitive Informatics, pp. 223–229 (2010)
18. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (2006)

Macro Features Based Text Categorization

Dandan Wang, Qingcai Chen, Xiaolong Wang, and Buzhou Tang

MOS-MS Key lab of NLP & Speech
Harbin Institute of Technology Shenzhen Graduate School
Shenzhen 518055, P.R. China
{wangdandanhit,qingcai.chen,tangbuzhou}@gmail.com,
wangxl@insun.hit.edu.cn

Abstract. Text Categorization (TC) is one of the key techniques in web information processing. A lot of approaches have been proposed to do TC; most of them are based on the text representation using the distributions and relationships of terms, few of them take the document level relationships into account. In this paper, the document level distributions and relationships are used as a novel type features for TC. We called them macro features to differentiate from term based features. Two methods are proposed for macro features extraction. The first one is semi-supervised method based on document clustering technique. The second one constructs the macro feature vector of a text using the centroid of each text category. Experiments conducted on standard corpora Reuters-21578 and 20-newsgroup, show that the proposed methods can bring great performance improvement by simply combining macro features with classical term based features.

Keywords: text categorization, text clustering, centroid-based classification, macro features.

1 Introduction

Text categorization (TC) is one of the key techniques in web information organization and processing [1]. The task of TC is to assign texts to predefined categories based on their contents automatically [2]. This process is generally divided into five parts: pre-processing, feature selection, feature weighting, classification and evaluation. Among them, feature selection is the key step for classifiers. In recent years, many popular feature selection approaches have been proposed, such as Document Frequency (DF), Information Gain (IG), Mutual Information (MI), $\chi 2$ Statistic (CHI) [1], Weighted Log Likelihood Ratio (WLLR) [3], Expected Cross Entropy (ECE) [4] etc. Meanwhile, feature clustering, a dimensionality reduction technique, has also been widely used to extract more sophisticated features [5-6]. It extracts new features of one type from auto-clustering results for basic text features. Baker (1998) and Slonim (2001) have proved that feature clustering is more efficient than traditional feature selection methods [5-6]. Feature clustering can be classified into supervised, semi-supervised and unsupervised feature clustering. Zheng (2005) has shown that the semi-supervised feature clustering can outperform other two type techniques [7]. However, once the performance of feature clustering is not very good, it may yield even worse results in TC.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 211–219, 2011.
© Springer-Verlag Berlin Heidelberg 2011

While the above techniques take term level text features into account, centroid-based classification explored text level relationships [8-9]. By this centroid-based classification method, each class is represented by a centroid vector. Guan (2009) had shown good performance of this method [8]. He also pointed out that the performance of this method is greatly affected by the weighting adjustment method. However, current centroid based classification methods do not use the text level relationship as a new type of text feature rather than treat the exploring of such relationship as a step of classification.

Inspired by the term clustering and centroid-based classification techniques, this paper introduces a new type of text features based on the mining of text level relationship. To differentiate from term level features, we call the text level features as macro features, and the term level features as micro features respectively. Two methods are proposed to mining text relationships. One is based on text clustering, the probability distribution of text classes in each cluster is calculated by the labeled class information of each sampled text, which is finally used to compose the macro features of each test text. Another way is the same technique as centroid based classification, but for a quite different purpose. After we get the centroid of each text category through labeled training texts, the macro features of a given testing text are extracted through the centroid vector of its nearest text category. For convenience, the macro feature extraction methods based on clustering and centroid are denoted as MFCl and MFCe respectively in the following content. For both macro feature extraction methods, the extracted features are finally combined with traditional micro features to form a unified feature vector, which is further input into the state of the art text classifiers to get text categorization result. It means that our centroid based macro feature extraction method is one part of feature extraction step, which is different from existing centroid based classification techniques.

This paper is organized as follows. Section 2 introduces macro feature extraction techniques used in this paper. Section 3 introduces the experimental setting and datasets used. Section 4 presents experimental results and performance analysis. The paper is closed with conclusion.

2 Macro Feature Extraction

2.1 Clustering Based Method MFCl

In this paper, we extract macro features by K-means clustering algorithm [10] which is used to find cluster centers iteratively. Fig 1 gives a simple sketch to demonstrate the main principle. In Fig 1, there are three categories denoted by different shapes: rotundity, triangle and square, while unlabeled documents are denoted by another shape. The unlabeled documents are distributed randomly. Cluster 1, Cluster 2, Cluster 3 are the cluster centers after clustering. For each test document t_i, we calculate the Euclidean distance between the test document r and each cluster center to get the nearest cluster. It is demonstrated that the Euclidean distance is 0.5, 0.7 and 0.9 respectively. t_i is nearest to Cluster 3. The class probability vector of the nearest cluster is selected as the macro feature of the test document. In Cluster 3, there are 2 squares, 2 rotundities and 7 triangles together. Therefore, we can know the macro feature vector of t_i equals to (7/11, 2/11, 2/11).

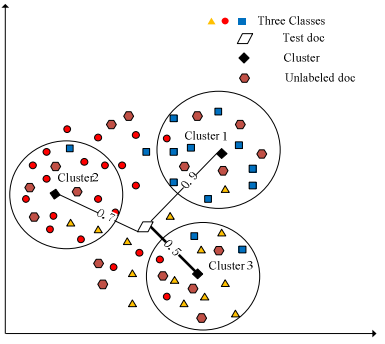


Fig. 1. Sketch of the MFCl

Algorithm 1. MFCl (Macro Features based on Clustering)

Consider an m-class classification problem with $m \geq 2$. There are n training samples $\{(x_1, y_1),(x_2, y_2),(x_3, y_3)...(x_n, y_n)\}$ with d dimensional feature vector $x_i \in \Re^n$ and corresponding class $y_i \in (1,2,3,...,m)$. MFCl can be shown as follows.

Input: The training data n

Output: Macro features

Procedure:

(1) K-means clustering. We set k as the predefined number of classes, that is m.

(2) Extraction of macro features. For each cluster, we obtain two vectors, one is the centroid vector CV which is the average of feature vectors of the documents belonging to the cluster, and the other is the class probability vector CPV which represents the probability of the clusters belonging to each class. For example, suppose cluster CL_j contains N_i labeled documents belonging to class y_i, then the class probability vector of the cluster CL_j can be described as:

$$CPV_j^c = (\frac{N_1}{\sum_{i=1}^{m} N_i}, \frac{N_2}{\sum_{i=1}^{m} N_i}, \frac{N_3}{\sum_{i=1}^{m} N_i}, ..., \frac{N_m}{\sum_{i=1}^{m} N_i}) \quad (1)$$

Where CPV_i^d represents the class probability vector of the cluster CL_j.

For each document D_i, we calculate the Euclidean distance between the document feature vector and the CV of each cluster. The class probability vector of the nearest cluster is selected as the macro features of the document if their distance metric reaches to a predefined minimal value of similarity, otherwise the macro features of the document will be set to a default value. As we have no prior information about the document, the default value is set based on the equal probability of belonging to each class, which is:

$$CPV_i^d = (\frac{1}{m}, \frac{1}{m}, \frac{1}{m}, ..., \frac{1}{m}) \quad (2)$$

Where CPV_i^d *represents the class probability vector of the document* D_i*. After obtaining the macro features of each document, we add those macro features to the micro feature vector space. Finally, each document is represented by a* $d+m$ *dimensional feature vector.*

$$FFV_i = (x_i, CPV_i^d) \tag{3}$$

Where FFV_i *represents the final feature vector of document* D_i

2.2 Centroid Based Method MFCe

In this paper, we extract macro features by Rocchio approach which assigns a centroid to each category by training set [11]. Fig 2 gives a simple sketch to demonstrate the main principle. In Fig 2, there are three categories denoted by different shapes: rotundity, triangle and square, while unlabeled documents are denoted by another shape. The unlabeled documents are distributed randomly. Cluster 1, Cluster 2, Cluster 3 are the cluster centers after clustering. For each test document t_i, we calculate the Euclidean distance between the test document and each cluster center to get the nearest cluster. It is demonstrated that the Euclidean distance is 0.5, 0.7 and 0.9 respectively. t_i is nearest to Cluster 3. The class probability vector of the nearest cluster is selected as the macro feature of the test document. In Cluster 3, there are 2 squares, 2 rotundities and 7 triangles together. Therefore, we can know the macro feature vector of t_i equals to (7/11, 2/11, 2/11).

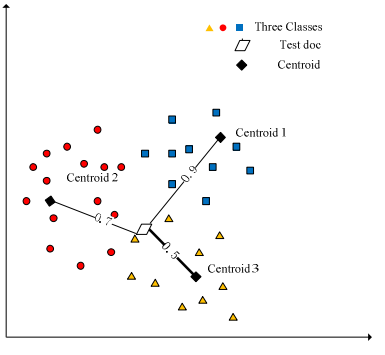


Fig. 2. Illustration of MFCe basic idea

Algorithm 2. MFCe (Macro Features based on Centroid Classification)
Here, the variables are the same as approach MFCl proposed in section 2.1.

Input: The training data n
Output: Macro features
Procedure:

(1) Partition the training corpus into two parts P_1 *and* P_2*.* P_1 *is used for the centroid-based classification,* P_2 *is used for Neural Network or SVM classification. Here, both* P_1 *and* P_2 *use the entire training corpus.*

(2) *Centroid-based classification. Rocchio algorithm is used for the centroid-based classification. After performing Rocchio algorithm, each centroid* j *in* P_1 *obtains a corresponding centroid vector* CV_j

(3) *Extraction of macro features. For each document* D_i *in* P_2*, we calculate the Euclidean distance between document* D_i *and each centroid in* P_1*, the vector of the nearest centroid is selected as the macro feature of document* D_i*. The macro feature is added to the micro feature vector of the document* D_i *for classification.*

---

3 Databases and Experimental Setting

3.1 Databases

Reuters-21578[1]**.** There are 21578 documents in this 52-category corpus after removing all unlabeled documents and documents with more than one class labels. Since the distribution of documents over the 52 categories is highly unbalanced, we only use the most populous 10 categories in our experiment [8]. A dataset containing 7289 documents with 10 categories are constructed. This dataset is randomly split into two parts: training set and testing set. The training set contains 5230 documents and the testing set contains 2059 documents. Clustering is performed only on the training set.

20-newsgroup[2]**.** The 20-newsgroup dataset is composed of 19997 articles single almost over 20 different Usenet discussion groups. This corpus is highly balanced. It is also randomly divided into two parts: 13296 documents for training and 6667 documents for testing. Clustering is also performed only on the training set.
For both corpora, **Lemur** is used for etyma extraction. IDF scores for feature weighting are extracted from the whole corpus. Stemming and stopping-word removal are applied.

3.2 Experimental Setting

Feature Selection. ECE is selected as the feature selection method in our experiment. 3000 dimensional features are selected out by this method.

Clustering. K-means method is used for clustering. K is set to be the number of class. In this paper, we have 10 and 20 classes for Reuters-21578 and 20-newsgroup respectively. When judging the nearest cluster of some document, the threshold of similarity is set to different values between 0 and 1 as needed. The best threshold of similarity for cluster judging is set to 0.45 and 0.54 for Reuters-21578 and 20-newsgroup respectively by a four-fold cross validation.

Classification. The parameters in Rocchio are set as follows: $\alpha = 0.5$, $\beta = 0.3$, $\gamma = 0.2$. SVM and Neural Network are used as classifiers. LibSVM[3] is used as the

---

[1] http://ronaldo.tcd.ie/esslli07/sw/step01.tgz
[2] http://people.csail.mit.edu/jrennie/20Newsgroups/
[3] LIBLINEAR:http://www.csie.ntu.edu.tw/~cjlin/liblinear/

tool of SVM classification where the linear kernel and the default settings are applied. For Neural Network in short for NN, three-layer structure with 50 hidden units and cross-entropy loss function is used. The inspiring function is sigmoid and linear, respectively, for the second and third layer.

In this paper, we use "MFCl+SVM" to denote the TC task conducted by inputting the combination of MFCl with traditional features into the SVM classifier. By the same way, we get four types of TC methods based on macro features, i.e., MFCl+SVM, MFCl+NN, MFCe+SVM and MFCe+NN. Moreover, macro and micro averaging F-measure denoted as macro-F1 and micro-F1 respectively are used for performance evaluation in our experiment.

4 Experimental Results

4.1 Performance Comparison of Different Methods

Several experiments are conducted with MFCl and MFCe. To provide a baseline for comparison, experiments are also conducted on Rocchio, SVM, Neural Network without using macro features. They are denoted as Rocchio, SVM and NN respectively. All these methods are using the same traditional features as those combined with MFCl and MFCe in macro features based experiments. The overall categorization results of these methods on both Reuters-21578 and 20-newsgroup are shown in Table 1.

Table 1. Overall TC Performance of MFC1 and MFCe

Classifier	Reuters-21578		20-newsgroup	
	macro-F1	micro-F1	macro-F1	micro-F1
SVM	0.8654	0.9184	0.8153	0.8155
NN	0.8498	0.9027	0.7963	0.8056
MFCl+SVM	**0.8722**	**0.9271**	**0.8213**	**0.8217**
MFCl+NN	0.8570	0.9125	0.8028	0.8140
Rocchio	0.8226	0.8893	0.7806	0.7997
MFCe+SVM	**0.8754**	**0.9340**	**0.8241**	**0.8239**
MFCe+NN	0.8634	0.9199	0.8067	0.8161

Table 1 shows that both the MFCl+SVM and MFCl+NN outperform the SVM and NN respectively on two datasets. On Reuters-21578, The improvement of macro-F1 and micro-F1 achieves about 0.79% and 0.95% respectively compared to SVM, and the improvement achieves about 0.85% and 1.09% respectively compared to Neural Network. On 20-newsgroup, the improvement of macro-F1 and micro-F1 achieves about 0.74% and 0.76% respectively compared to SVM, and the improvement achieves about 0.82% and 1.04% respectively compared to Neural Network.

Furthermore, Table 1 demonstrates that SVM with MFCe and NN with MFCe outperform the separated SVM and NN respectively on both two standard datasets. They all perform better than separated centroid-based classification Rocchio. Thereinto NN with MFCe can achieve the most about 1.91% and 1.60% improvement respectively comparing with separated NN on micro-F1 and macro-F1 on Reuters-21578. Both the training set for centroid-based classification and for SVM or NN classification use all of the training set.

4.2 Effectiveness of Labeled Data in MFCl

In Fig 3 and 4, we demonstrate the effect of different sizes of labeled set on micro-F1 for Reuters-21578 and 20-newsgroup using MFCl on SVM and NN.

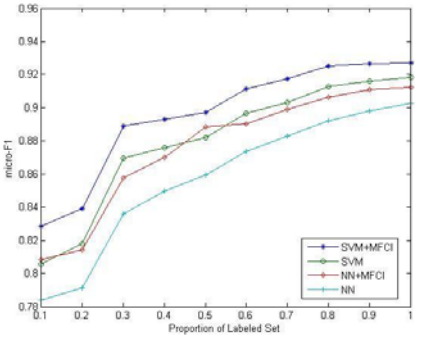


Fig. 3. Performance of different sizes of labeled data using for MFCl training on Reuters-21578

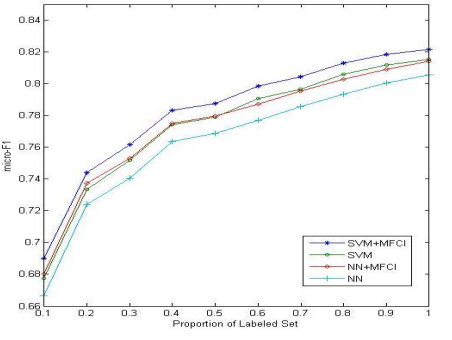


Fig. 4. Performance of different sizes of labeled data using for MFCl training on 20-newsgroup

These figures show that the performance gain drops as the size of the labeled set increases on both two standard datasets. But it still gets some performance gain as the proportion of the labeled set reaches up to 100%. On Reuters-21578, it gets approximately 0.95% and 1.09% gain respectively for SVM and NN, and the performance gain is 0.76% and 0.84% respectively for SVM and NN on 20-newsgroup.

4.3 Effectiveness of Labeled Data in MFCe

In Table 2 and 3, we demonstrate the effect of different sizes of labeled set on micro-F1 for the Reuters-21578 and 20-newsgroup dataset.

Table 2. Micro-F1 of using different sizes of labeled set for MFCe training on Reuters-21578

Reuters-21578				
labeled set (%)	SVM+MFCe	SVM	NN+MFCe	NN
10	0.8107	0.8055	0.7899	0.7841
20	0.8253	0.8182	0.7992	0.7911
30	0.8785	0.8696	0.8455	0.8358
40	0.8870	0.8758	0.8620	0.8498
50	0.8946	0.8818	0.8725	0.8594
60	0.9109	0.8967	0.8879	0.8735
70	0.9178	0.9032	0.8991	0.8831
80	0.9283	0.913	0.9087	0.8919
90	0.9316	0.9162	0.9150	0.8979
100	0.9340	0.9184	0.9199	0.9027

Table 3. Micro-F1 of using different sizes of labeled set for MFCe training on 20-newsgroup

20-newsgroup				
labeled set (%)	SVM+MFCe	SVM	NN+MFCe	NN
10	0.6795	0.6774	0.6712	0.6663
20	0.7369	0.7334	0.7302	0.7241
30	0.7562	0.7519	0.7478	0.7407
40	0.7792	0.7742	0.7713	0.7635
50	0.7842	0.7788	0.7768	0.7686
60	0.7965	0.7905	0.7856	0.7768
70	0.8031	0.7967	0.7953	0.7857
80	0.8131	0.8058	0.8034	0.7935
90	0.8197	0.8118	0.8105	0.8003
100	0.8239	0.8155	0.8161	0.8056

These tables show that the gain rises as the size of the labeled set increases on both two standard datasets. On Reuters-21578, it gets approximately 1.70% and 1.90% gain respectively for SVM and NN when the proportion of the size of the labeled set reaches up to 100%. On 20-newsgroup, the gain is about 1.03% and 1.30% respectively for SVM and NN.

4.4 Comparison of MFCl and MFCe

In Fig 5 and 6, we demonstrate the differences of performance between SVM+MFCe (NN+MFCe) and SVM+MFCl (NN+MFCl) on Reuters-21578 and 20-newsgroup.

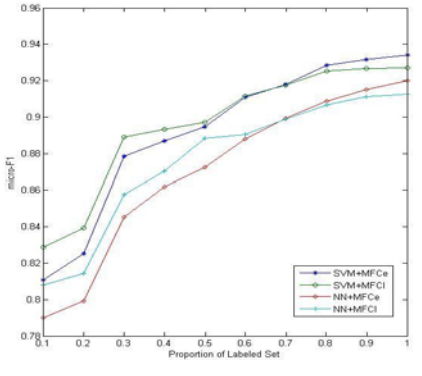


Fig. 5. Comparison of MFCl and MFCe with proportions of labeled data on Reuters-21578

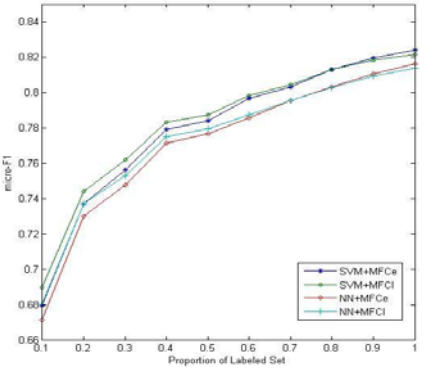


Fig. 6. Comparison of MFCl and MFCe with proportions of labeled data on 20-newsgroup

These graphs show that SVM+MFCl (NN+MFCl) outperforms SVM+MFCe (NN+MFCe) when the proportion of the labeled set is less than approximately 70% for Reuters-21578, and 80% for 20-newgroup. As the proportion increasingly reaches up to this point, SVM+MFCe (NN+MFCe) gets better than SVM+MFCl (NN+MFCl).

It can be explained the MFCl algorithm is dependent on labeled set and the unlabeled set, while the MFCe algorithm is dependent only on the labeled set. When the proportion of the labeled set is small, the MFCl algorithm can benefit more from the unlabeled set than the MFCe algorithm. As the proportion of the labeled set increases, the benefits of unlabeled data for MFCl algorithm drop. Finally MFCl performs worse than MFCe after the proportion of labeled data greater than 70%.

5 Conclusion

In this paper, two macro feature extraction methods, i.e., MFCl and MFCe are proposed to enhance text categorization performance. The MFCl uses the probability of clusters belonging to each class as the macro features, while the MFCe combines the centroid-based classification with traditional classifiers like SVM or Neural Network. Experiments conducted on Reuters-21578 and 20-newsgroup show that combining macro features with traditional micro features achieved promising improvement on micro-F1 and macro-F1 for both macro feature extraction methods.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China (No. 60973076).

References

1. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: International Conference on Machine Learning (1997)
2. Yang, Y.: An Evaluation of Statistical Approaches to Text Categorization. Journal of Information Retrieval 1, 69–90 (1999)
3. Li, S., Xia, R., Zong, C., Huang, C.-R.: A Framework of Feature Selection Methods for Text Categorization. In: International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, pp. 692–700 (2009)
4. How, B.C., Narayanan, K.: An Empirical Study of Feature Selection for Text Categorization based on Term Weightage. In: International Conference on Web Intelligence, pp. 599–602 (2004)
5. Baker, L.D., McCallumlt, A.K.: Distributional Clustering of Words for Text Classification. In: ACM Special Inspector General for Iraq Reconstruction Conference on Research and Development in Information Retrieval, pp. 96–103 (1998)
6. Slonim, N., Tishby, N.: The Power of Word Clusters for Text Classification. In: European Conference on Information Retrieval (2001)
7. Niu, Z.-Y., Ji, D.-H., Tan, C.L.: A Semi-Supervised Feature Clustering Algorithm with Application toWord Sense Disambiguation. In: Human Language Technology Conference and Conference on Empirical Methods in Natural Language, pp. 907–914 (2005)
8. Guan, H., Zhou, J., Guo, M.: A Class-Feature-Centroid Classifier for Text Categorization. In: World Wide Web Conference, pp. 201–210 (2009)
9. Tan, S., Cheng, X.: Using Hypothesis Margin to Boost Centroid Text Classifier. In: ACM Symposium on Applied Computing, pp. 398–403 (2007)
10. Khan, S.S., Ahmad, A.: Cluster center initialization algorithm for K-means clustering. Pattern Recognition Letters 25, 1293–1302 (2004)
11. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1), 1–47 (2002)

Univariate Marginal Distribution Algorithm in Combination with Extremal Optimization (EO, GEO)

Mitra Hashemi[1] and Mohammad Reza Meybodi[2]

[1] Department of Computer Engineering and Information Technology,
Islamic Azad University Qazvin Branch, Qazvin, Iran
Mitra.hash@yahoo.com
[2]Department of Computer Engineering and Information Technology,
Amirkabir University of Technology, Tehran, Iran
mmeybodi@aut.ac.ir

Abstract. The UMDA algorithm is a type of Estimation of Distribution Algorithms. This algorithm has better performance compared to others such as genetic algorithm in terms of speed, memory consumption and accuracy of solutions. It can explore unknown parts of search space well. It uses a probability vector and individuals of the population are created through the sampling. Furthermore, EO algorithm is suitable for local search of near global best solution in search space, and it dose not stuck in local optimum. Hence, combining these two algorithms is able to create interaction between two fundamental concepts in evolutionary algorithms, exploration and exploitation, and achieve better results of this paper represent the performance of the proposed algorithm on two NP-hard problems, multi processor scheduling problem and graph bi-partitioning problem.

Keywords: Univariate Marginal Distribution Algorithm, Extremal Optimization, Generalized Extremal Optimization, Estimation of Distribution Algorithm.

1 Introduction

During the ninetieth century, Genetic Algorithms (GAs) helped us solve many real combinatorial optimization problems. But the deceptive problem where performance of GAs is very poor has encouraged research on new optimization algorithms. To combat these dilemma some researches have recently suggested Estimation of Distribution Algorithms (EDAs) as a family of new algorithms [1, 2, 3]. Introduced by Muhlenbein and Paaβ, EDAs constitute an example of stochastic heuristics based on populations of individuals each of which encodes a possible solution of the optimization problem. These populations evolve in successive generations as the search progresses–organized in the same way as most evolutionary computation heuristics. This method has many advantages which can be illustrated by avoiding premature convergence and use of a compact and short representation.

In 1996, Muhlenbein and PaaB [1, 2] have proposed the Univariate Marginal Distributions Algorithm (UMDA), which approximates the simple genetic algorithm.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 220–227, 2011.
© Springer-Verlag Berlin Heidelberg 2011

One problem of GA is that it is very difficult to quantify and thus analyze these effects. UMDA is based on probability theory, and its behavior can be analyzed mathematically.

Self-organized criticality has been used to explain behavior of complex systems in such different areas as geology, economy and biology. To show that SOC [5,6] could explain features of systems like the natural evolution, Bak and Sneepen developed a simplified model of an ecosystem to each species, a fitness number is assigned randomly, with uniform distribution, in the range [0,1]. The least adapted species, one with the least fitness, is then forced to mutate, and a new random number assigned to it. In order to make the Extremal Optimization (EO) [8,9] method applicable to a broad class of design optimization problems, without concern to how fitness of the design variables would be assigned, a generalization of the EO, called Generalized Extremal Optimization (GEO), was devised. In this new algorithm, the fitness assignment is not done directly to the design variables, but to a "population of species" that encodes the variables.

The ability of EO in exploring search space was not as well as its ability in exploiting whole search space; therefore combination of two methods, UMDA and EO/GEO(UMDA-EO, UMDA-GEO) , could be very useful in exploring unknown area of search space and also for exploiting the area of near global optimum.

This paper has been organized in five major sections: section 2 briefly introduces UMDA algorithm; in section 3, EO and GEO algorithms will be discussed; in section 4 suggested algorithms will be introduced; section 5 contains experimental results; finally, section 6 which is the conclusion

2 Univariate Marginal Distribution Algorithm

The Muhlenbein introduced UMDA [1,2,12] as the simplest version of estimation of distribution algorithms (EDAs). SUMDA starts from the *central probability vector* that has value of 0.5 for each locus and falls in the central point of the search space. Sampling this probability vector creates random solutions because the probability of creating a 1 or 0 on each locus is equal. Without loss of generality, a binary-encoded solution $x=(x_1,...,x_l)\in \{0,1\}^l$ is sampled from a probability vector p(t). At iteration t, a population S(t) of n individuals are sampled from the probability vector p(t). The samples are evaluated and an interim population D(t) is formed by selecting μ (μ<n) best individuals. Then the probability vector is updated by extracting statistics information from D(t) as follows:

$$p'(t) = \frac{1}{\mu}\sum_{k=1}^{k=\mu} x_k(t) \tag{1}$$

The mutation operation always changes locus i={1,…,l}, if a random number r=rand(0,1)< p_m (p_m is the mutation probability), then mutate p(i,t) using the following formula:

$$p'(i,t) = \begin{cases} p(i,t)*(1.0-\delta_m), p(i,t) > 0.5 \\ p(i,t), p(i,t) = 0.5 \\ p(i,t)*(1.0-\delta_m)+\delta_m, p(i,t) < 0.5 \end{cases} \tag{2}$$

Where δ_m is mutation shift. After the mutation operation, a new set of samples is generated by the new probability vector and this cycle is repeated.

As the search progresses, the elements in the probability vector move away from their initial settings of 0.5 towards either 0.0 or 1.0, representing samples of height fitness. The search stops when some termination condition holds, e.g., the maximum allowable number of iterations $t_{\max}$ is reached.

3 Extremal Optimization Algorithm

Extremal optimization [4,8,9] was recently proposed by Boettcher and Percus. The search process of EO eliminates components having extremely undesirable (worst) performance in sub-optimal solution, and replaces them with randomly selected new components iteratively. The basic algorithm operates on a single solution S, which usually consists of a number of variables $x_i (1 \le i \le n)$. At each update step, the variable x_i with worst fitness is identified to alter. To improve the results and avoid the possible dead ends, Boettcher and Percus subsequently proposed τ -EO that is regarded as a general modification of EO by introducing a parameter. All variables x_i are ranked according to the relevant fitness. Then each independent variable x_i to be moved is selected according to the probability distribution (3).

$$p = k^{-\tau} \tag{3}$$

Sousa and Ramos have proposed a generalization of the EO that was named the Generalized Extremal Optimization (GEO) [10] method. To each species (bit) is assigned a fitness number that is proportional to the gain (or loss) the objective function value has in mutating (flipping) the bit. All bits are then ranked. A bit is then chosen to mutate according to the probability distribution. This process is repeated until a given stopping criteria is reached .

4 Suggested Algorithm

We combined UMDA with EO for better performance. Power EO is less in comparison with other algorithms like UMDA in exploring whole search space thus with combination we use exploring power of UMDA and exploiting power of EO in order to find the best global solution, accurately.

We select the best individual in part of the search space, and try to optimize the best solution on the population and apply a local search in landscape, most qualified person earns and we use it in probability vector learning process.

According to the subjects described, the overall shape of proposed algorithms (UMDA-EO, UMDA-GEO) will be as follow:

1. Initialization
2. Initialize probability vector with 0.5
3. Sampling of population with probability vector

4. Matching each individual with the issue conditions (equal number of nodes in both parts)
    a. Calculate the difference between internal and external (D) cost for all nodes
    b. If A> B transport nodes with more D from part A to part B
    c. If B> A transport nodes with more D from part A to part B
    d. Repeat steps until achieve an equal number of nodes in both
5. Evaluation of population individuals
6. Replace the worst individual with the best individual population (elite) of the previous population
7. Improve the best individual in the population using *internal EO (internal GEO)*, and injecting to the population
8. Select μ best individuals to form a temporary population
9. Making a probability vector based on temporary population according (1)
10. Mutate in probability vector according (2)
11. Repeat steps from step 3 until the algorithm stops

Internal EO:
1. Calculate fitness of solution components
2. Sort solution components based on fitness as ascent
3. Choose one of components using the (3)
4. Select the new value for exchange component according to the problem
5. Replace new value in exchange component and produce a new solution
6. Repeat from step1 until there are improvements.

Internal GEO:
1. Produce children of current solution and calculate their fitness
2. Sort solution components based on fitness as ascent
3. Choose one of the children as a current solution according to (3)
4. Repeat the steps until there are improvements.

Results on both benchmark problems represent performance of proposed algorithms.

5 Experiments and Results

To evaluate the efficiency of the suggested algorithm and in order to compare it with other methods two NP-hard problem, Multi Processor Scheduling problem and Graph Bi-partitioning problem are used. The objective of scheduling is usually to minimize the completion time of a parallel application consisted of a number of tasks executed in a parallel system. Samples of problems that the algorithms used to compare the performance can be found in reference [11]. Graph bi-partitioning problem consists of dividing the set of its nodes into two disjoint subsets containing equal number of nodes in such a way that the number of graph edges connecting nodes belonging to different subsets (i.e., the cut size of the partition) are minimized. Samples of problems that the algorithms used to compare the performance can be found in reference [7].

5.1 Graph Bi-partitioning Problem

We use bit string representation to solve this problem. 0 and 1 in this string represent two separate part of graph. Also in order to implement EO for this problem, we use [8] and [9]. These references use initial clustering. In this method to compute fitness of each component, we use ratio of neighboring nodes in each node for matching each individual with the issue conditions (equal number of nodes in both parts), using KL algorithm [12].

In the present study, we set parameters using calculate relative error in different runs. Suitable values for this parameters are as follow: mutation probability (0.02), mutation shift (0.2), population size (60), temporary population size (20) and maximum iteration number is 100.

In order to compare performance of methods, UMDA-EO, EO-LA and EO, We set τ =1.8 that is best value for EO algorithm based on calculating mean relative error in 10 runs. Fig.1 shows the results and best value for τ parameter.

The algorithms compare UMDA-EO, EO-LA and τ-EO and see the change effects; the parameter value τ for all experiments is 1.8.

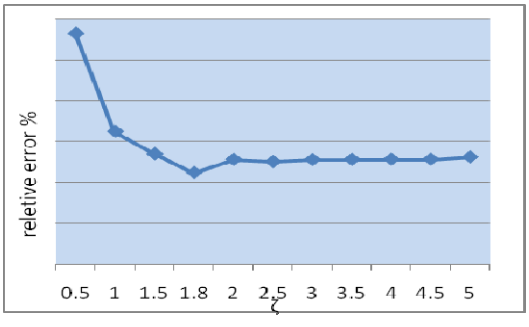


Fig. 1. Select best value for τ parameter

Table 3 shows results of comparing algorithms for this problem. We observe the proposed algorithm in most of instances has minimum and best value in comparing with other algorithms.

Comparative study of algorithms for solving the graph bi-partitioning problem is used instances that stated in the previous section. Statistical analysis solutions produced by these algorithms are shown in Table 3. As can be UMDA-EO algorithm in almost all cases are better than rest of the algorithms. Compared with EO-LA (EO combined with learning automata) can be able to improve act of exploiting near areas of suboptimal solutions but do not explore whole search space well.

Fig.2 also indicates that average error in samples of graph bi-partitioning problem in suggested algorithm is less than other algorithms.

Good results of the algorithm are because of the benefits of both algorithms and elimination of the defects. UMDA algorithm emphasizes at searching unknown areas in space, and the EO algorithm using previous experiences and the search near the global optimum locations and find optimal solution.

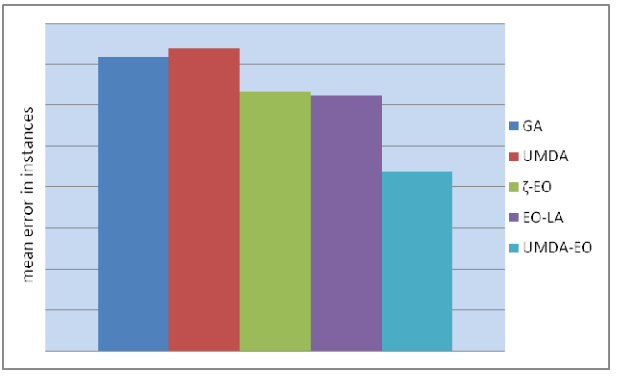


Fig. 2. Comparison mean error in UMDA-EO with other methods

5.2 Multiprocessor Scheduling Problems

We use [10] for implementation of UMDA-GEO in multiprocessor scheduling problem. Samples of problems that the algorithms used to compare the performance have been addressed in reference [11]. In this paper multiprocessor scheduling with priority and without priority is discussed. We assume 50 and 100 task in parallel system with 2,4,8 and 16 processor. Complete description about representation and etc. are discussed by P. Switalski and F. Seredynski [10].

We set parameter using calculate relative error in different runs; suitable values for this parameter are as follow: mutation probability (0.02), mutation shift (0.05), pop size (60), temporary pop size (20) and maximum iteration number is 100.

To compare performance of methods, UMDA-GEO, GEO, We set τ =1.2; this is best value for EO algorithm based on calculating mean relative error in 10 runs.

In order to compare the algorithms in solving scheduling problem, each of these algorithms runs 10 numbers and minimum values of results are presented in Tables 1 and 2. In this comparison, value of τ parameter is 1.2. Results are in two style of implementation, with and without priority.

Results in Tables 1 and 2 represent in almost all cases proposed algorithm (UMDA-GEO) had better performance and shortest possible response time. When number of processor is few most of algorithms achieve the best response time, but when numbers of processors are more advantages of proposed algorithm are considerable.

Table 1. Results of scheduling with 50 tasks

processor	GA	UMDA	GEO	UMDA-GEO
		Without priority		
2	131	131	131	131
4	66	33	33	33
8	35	17	18	16
16	21	12	10	10
		With priority		
2	131	132	131	131
4	78	74	66	68
8	57	55	55	55
16	55	55	55	55

Table 2. Results of scheduling with 50 tasks

processsor	GA	UMDA	GEO	UMDA-GEO
		Without priority		
2	291	291	291	291
4	146	75	72	72
8	77	24	24	23
16	23	23	23	22
		With priority		
2	291	291	291	291
4	189	163	146	148
8	153	108	98	94
16	150	99	97	94

Table 3. Experimental results of graph bi-partitioning problem

graph	GA	UMDA	ζ-EO	EO-LA	UMDA-EO
3elt	335.5	351	330.5	327	277.5
4elt	772	814.5	728	722.5	748.5
Add20	4287	4651	4942	4683	3222
Add32	2179.5	2403	254.5	216	502.5
Crack	445	428.5	380	378.5	385
CS4	1016	951	769	790	615
CTI	2428.5	2389	1747	1797	1479
Data	510	495	672	674	495
FE-4elt2	196	210.5	494	502	197
FE-sphere	665	666.5	677	678	654
UK	67	67	35	31	39
Whitaker3	224.5	218	442	453	207.5
Wing-nodal	9839	9887	9287	9263.5	6816

6 Conclusion

Findings of the present study implies that, the suggested algorithm (UMDA-EO and UMDA-GEO) has a good performance in real-world problems, multiprocessor scheduling problem and graph bi-partitioning problem. They combine the two methods and both benefits that were discussed in the paper and create a balance between two concepts of evolutionary algorithms, exploration and exploitation.

UMDA acts in the discovery of unknown parts of search space and EO search near optimal parts of landscape to find global optimal solution; therefore, with combination of two methods can find global optimal solution accurately.

References

1. Yang, S.: Explicit Memory scheme for Evolutionary Algorithms in Dynamic Environments. SCI, vol. 51, pp. 3–28. Springer, Heidelberg (2007)
2. Tianshi, C., Tang, K., Guoliang, C., Yao, X.: Analysis of Computational Time of Simple Estimation of Distribution Algorithms. IEEE Trans. Evolutionary Computation 14(1) (2010)

3. Hons, R.: Estimation of Distribution Algorithms and Minimum Relative Entropy, phd. Thesis. university of Bonn (2005)
4. Boettcher, S., Percus, A.G.: Extremal Optimization: An Evolutionary Local-Search Algorithm, http://arxiv.org/abs/cs.NE/0209030
5. http://en.wikipedia.org/wiki/Self-organized_criticality
6. Bak, P., Tang, C., Wiesenfeld, K.: Self-organized Criticality. Physical Review A 38(1) (1988)
7. http://staffweb.cms.gre.ac.uk/~c.walshaw/partition
8. Boettcher, S.: Extremal Optimization of Graph Partitioning at the Percolation Threshold. Physics A 32(28), 5201–5211 (1999)
9. Boettcher, S., Percus, A.G.: Extremal Optimization for Graph Partitioning. Physical Review E 64, 21114 (2001)
10. Switalski, P., Seredynski, F.: Solving multiprocessor scheduling problem with GEO metaheuristic. In: IEEE International Symposium on Parallel&Distributed Processing (2009)
11. http://www.kasahara.elec.waseda.ac.jp
12. Mühlenbein, H., Mahnig, T.: Evolutionary Optimization and the Estimation of Search Distributions with Applications to Graph Bipartitioning. Journal of Approximate Reasoning 31 (2002)

Promoting Diversity in Particle Swarm Optimization to Solve Multimodal Problems*

Shi Cheng[1,2], Yuhui Shi[2], and Quande Qin[3]

[1] Department of Electrical Engineering and Electronics,
University of Liverpool, Liverpool, UK
shi.cheng@liverpool.ac.uk
[2] Department of Electrical & Electronic Engineering,
Xi'an Jiaotong-Liverpool University, Suzhou, China
yuhui.shi@xjtlu.edu.cn
[3] College of Management, Shenzhen University, Shenzhen, China
qinquande@gmail.com

Abstract. Promoting diversity is an effective way to prevent premature converge in solving multimodal problems using Particle Swarm Optimization (PSO). Based on the idea of increasing possibility of particles "jump out" of local optima, while keeping the ability of algorithm finding "good enough" solution, two methods are utilized to promote PSO's diversity in this paper. PSO population diversity measurements, which include position diversity, velocity diversity and cognitive diversity on standard PSO and PSO with diversity promotion, are discussed and compared. Through this measurement, useful information of search in exploration or exploitation state can be obtained.

Keywords: Particle swarm optimization, population diversity, diversity promotion, exploration/exploitation, multimodal problems.

1 Introduction

Particle Swarm Optimization (PSO) was introduced by Eberhart and Kennedy in 1995 [6,9]. It is a population-based stochastic algorithm modeled on the social behaviors observed in flocking birds. Each particle, which represents a solution, flies through the search space with a velocity that is dynamically adjusted according to its own and its companion's historical behaviors. The particles tend to fly toward better search areas over the course of the search process [7].

Optimization, in general, is concerned with finding "best available" solution(s) for a given problem. For optimization problems, it can be simply divided into unimodal problem and multimodal problem. As the name indicated, a unimodal problem has only one optimum solution; on the contrary, multimodal problems have several or numerous optimum solutions, of which many are local optimal

* The authors' work was supported by National Natural Science Foundation of China under grant No. 60975080, and Suzhou Science and Technology Project under Grant No. SYJG0919.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 228–237, 2011.
© Springer-Verlag Berlin Heidelberg 2011

solutions. Evolutionary optimization algorithms are generally difficult to find the global optimum solutions for multimodal problems due to premature converge. Avoiding premature converge is important in multimodal problem optimization, i.e., an algorithm should have a balance between fast converge speed and the ability of "jump out" of local optima.

Many approaches have been introduced to avoid premature convergence [1]. However, these methods did not incorporate an effective way to measure the exploration/exploitation of particles. PSO with re-initialization, which is an effective way to promoting diversity, is utilized in this study to increase possibility for particles to "jump out" of local optima, and to keep the ability for algorithm to find "good enough" solution. The results show that PSO with elitist re-initialization has better performance than standard PSO. PSO population diversity measurements, which include position diversity, velocity diversity and cognitive diversity on standard PSO and PSO with diversity promotion, are discussed and compared. Through this measurement, useful information of search in exploration or exploitation state can be obtained.

In this paper, the basic PSO algorithm, and the definition of population diversity are reviewed in Section 2. In Section 3, two mechanisms for promoting diversity are utilized and described. The experiments are conducted in Section 4, which includes the test functions used, optimizer configurations, and results. Section 5 analyzes the population diversity of standard PSO and PSO with diversity promotion. Finally, Section 6 concludes with some remarks and future research directions.

2 Preliminaries

2.1 Particle Swarm Optimization

The original PSO algorithm is simple in concept and easy in implementation [10,8]. The basic equations are as follow:

$$v_{ij} = wv_{ij} + c_1\text{rand}()(p_i - x_{ij}) + c_2\text{Rand}()(p_n - x_{ij}) \quad (1)$$

$$x_{ij} = x_{ij} + v_{ij} \quad (2)$$

where w denotes the inertia weight and is less than 1, c_1 and c_2 are two positive acceleration constants, rand() and Rand() are functions to generate uniformly distributed random numbers in the range $[0, 1]$, v_{ij} and x_{ij} represent the velocity and position of the ith particle at the jth dimension, p_i refers to the best position found by the ith particle, and p_n refers to the position found by the member of its neighborhood that had the best fitness evaluation value so far.

Different topology structure can be utilized in PSO, which will have different strategy to share search information for every particle. Global star and local ring are two most commonly used structures. A PSO with global star structure, where all particles are connected to each other, has the smallest average distance in swarm, and on the contrary, a PSO with local ring structure, where every particle is connected to two near particles, has the biggest average distance in swarm [11].

2.2 Population Diversity Definition

The most important factor affecting an optimization algorithm's performance is its ability of "exploration" and "exploitation". Exploration means the ability of a search algorithm to explore different areas of the search space in order to have high probability to find good optimum. Exploitation, on the other hand, means the ability to concentrate the search around a promising region in order to refine a candidate solution. A good optimization algorithm should optimally balance the two conflicted objectives.

Population diversity of PSO is useful for measuring and dynamically adjusting algorithm's ability of exploration or exploitation accordingly. Shi and Eberhart gave three definitions on population diversity, which are position diversity, velocity diversity, and cognitive diversity [12,13]. Position, velocity, and cognitive diversity is used to measure the distribution of particles' current positions, current velocities, and *pbest*s (the best position found so far for each particles), respectively. Cheng and Shi introduced the modified definitions of the three diversity measures based on L_1 norm [3,4].

From diversity measurements, the useful information can be obtained. For the purpose of generality and clarity, m represents the number of particles and n the number of dimensions. Each particle is represented as x_{ij}, i represents the ith particle, $i = 1, \cdots, m$, and j is the jth dimension, $j = 1, \cdots, n$. The detailed definitions of PSO population diversities are as follow:

Position Diversity. Position diversity measures distribution of particles' current positions. Particles going to diverge or converge, i.e., swarm dynamics can be reflected from this measurement. Position diversity gives the current position distribution information of particles. Definition of position diversity, which based on the L_1 norm, is as follows

$$\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m} x_{ij} \qquad \mathbf{D}^p = \frac{1}{m}\sum_{i=1}^{m} |x_{ij} - \bar{x}_j| \qquad D^p = \frac{1}{n}\sum_{j=1}^{n} D_j^p$$

where $\bar{\mathbf{x}} = [\bar{x}_1, \cdots, \bar{x}_j, \cdots, \bar{x}_n]$, $\bar{\mathbf{x}}$ represents the mean of particles' current positions on each dimension. $\mathbf{D}^p = [D_1^p, \cdots, D_j^p, \cdots, D_n^p]$, which measures particles' position diversity based on L_1 norm for each dimension. D^p measures the whole swarm's population diversity.

Velocity Diversity. Velocity diversity, which gives the dynamic information of particles, measures the distribution of particles' current velocities, In other words, velocity diversity measures the "activity" information of particles. Based on the measurement of velocity diversity, particle's tendency of expansion or convergence could be revealed. Velocity diversity based on L_1 norm is defined as follows

$$\bar{\mathbf{v}} = \frac{1}{m}\sum_{i=1}^{m} v_{ij} \qquad \mathbf{D}^v = \frac{1}{m}\sum_{i=1}^{m} |v_{ij} - \bar{v}_j| \qquad D^v = \frac{1}{n}\sum_{j=1}^{n} D_j^v$$

where $\bar{\mathbf{v}} = [\bar{v}_1, \cdots, \bar{v}_j, \cdots, \bar{v}_n]$, $\bar{\mathbf{v}}$ represents the mean of particles' current velocities on each dimension; and $\mathbf{D}^v = [D_1^v, \cdots, D_j^v, \cdots, D_n^v]$, $\mathbf{D}^v$ measures velocity

diversity of all particles on each dimension. D^v represents the whole swarm's velocity diversity.

Cognitive Diversity. Cognitive diversity measures the distribution of *pbest*s for all particles. The measurement definition of cognitive diversity is the same as that of the position diversity except that it utilizes each particle's current personal best position instead of current position. The definition of PSO cognitive diversity is as follows

$$\bar{\mathbf{p}} = \frac{1}{m}\sum_{i=1}^{m} p_{ij} \qquad \mathbf{D}_j^c = \frac{1}{m}\sum_{i=1}^{m} |p_{ij} - \bar{p}_j| \qquad D^c = \frac{1}{n}\sum_{j=1}^{n} D_j^c$$

where $\bar{\mathbf{p}} = [\bar{p}_1, \cdots, \bar{p}_j, \cdots, \bar{p}_n]$ and $\bar{\mathbf{p}}$ represents the average of all particles' personal best position in history (*pbest*) on each dimension; $\mathbf{D}^c = [D_1^p, \cdots, D_j^p, \cdots, D_n^p]$, which represents the particles' cognitive diversity for each dimension based on L_1 norm. D^c measures the whole swarm's cognitive diversity.

3 Diversity Promotion

Population diversity is a measurement of population state in exploration or exploitation. It illustrates the information of particles' position, velocity, and cognitive. Particles diverging means that the search is in an exploration state, on the contrary, particles clustering tightly means that the search is in an exploitation state.

Particles re-initialization is an effective way to promote diversity. The idea behind the re-initialization is to increase possibility for particles "jump out" of local optima, and to keep the ability for algorithm to find "good enough" solution. Algorithm 1 below gives the pseudocode of the PSO with re-initialization. After several iterations, part of particles re-initialized its position and velocity in whole search space, which increased the possibility of particles "jump out" of local optima [5]. According to the way of keeping some particles, this mechanism can be divided into two kinds.

Random Re-initialize Particles. As its name indicates, random re-initialization means reserves particles by random. This approach can obtain a great ability of exploration due to the possibility that most of particles will have the chance to be re-initialized.

Elitist Re-initialize Particles. On the contrary, elitist re-initialization keeps particles with better fitness value. Algorithm increases the ability of exploration due to the re-initialization of worse preferred particles in whole search space, and at the same time, the attraction to particles with better fitness values.

The number of reserved particles can be a constant or a fuzzy increasing number, different parameter settings are tested in next section.

4 Experimental Study

Wolpert and Macerady have proved that under certain assumptions no algorithm is better than other one on average for all problems [14]. The aim of the

Algorithm 1. Diversity promotion in particle swarm optimization

```
Initialize velocity and position randomly for each particle in every dimension
while not found the "good" solution or not reaches the maximum iteration do
    Calculate each particle's fitness value
    Compare fitness value between current value and best position in history (personal best, termed as pbest). For each particle, if fitness value of current position is better than pbest, then update pbest as current position.
    Selection a particle which has the best fitness value from current particle's neighborhood, this particle is called the neighborhood best (termed as nbest).
    for each particle do
        Update particle's velocity according equation (1)
        Update particle's position according equation (2)
        Keep some particles' (α percent) position and velocity, re-initialize others randomly after each β iteration.
    end for
end while
```

experiment is not to compare the ability or the efficacy of PSO algorithm with different parameter setting or structure, but the ability to "jump out" of local optima, i.e., the ability of exploration.

4.1 Benchmark Test Functions and Parameter Setting

The experiments have been conducted on testing the benchmark functions listed in Table 1. Without loss of generality, seven standard multimodal test functions were selected, namely Generalized Rosenbrock, Generalized Schwefel's Problem 2.26, Generalized Rastrigin, Noncontinuous Rastrigin, Ackley, Griewank, and Generalized Penalized [15]. All functions are run 50 times to ensure a reasonable statistical result necessary to compare the different approaches, and random shift of the location of optimum is utilized in dimensions at each time.

In all experiments, PSO has 50 particles, and parameters are set as the standard PSO, let $w = 0.72984$, and $c_1 = c_2 = 1.496172$ [2]. Each algorithm runs 50 times, 10000 iterations in every run.

Due to the limit of space, the simulation results of three representative benchmark functions are reported here, which are Generalized Rosenbrock (f_1), Noncontinuous Rastrigin(f_4), and Generalized Penalized(f_7).

4.2 Experimental Results

As we are interested in finding an optimizer that will not be easily deceived by local optima, we use three measures of performance. The first is the best fitness value attained after a fixed number of iterations. In our case, we report the best result found after $10,000$ iterations. The second and the last are the middle and mean value of best fitness values in each run. It is possible that an algorithm will rapidly reach a relatively good result while become trapped onto a local optimum. These two values give a measure of the ability of exploration.

Table 1. The benchmark functions used in our experimental study, where n is the dimension of each problem, $\mathbf{z} = (\mathbf{x} - \mathbf{o})$, o_i is an randomly generated number in problem's search space S and it is different in each dimension, global optimum $\mathbf{x}^* = \mathbf{o}$, $f_{\min}$ is the minimum value of the function, and $S \subseteq \mathbb{R}^n$

Function	Test Function	n	S	$f_{\min}$
Rosenbrock	$f_1(\mathbf{x}) = \sum_{i=1}^{n-1}[100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2]$	100	$[-10, 10]^n$	-450.0
Schwefel	$f_2(\mathbf{x}) = \sum_{i=1}^{n} -z_i \sin(\sqrt{\lvert z_i \rvert}) + 418.9829n$	100	$[-500, 500]^n$	-330.0
Rastrigin	$f_3(\mathbf{x}) = \sum_{i=1}^{n}[z_i^2 - 10\cos(2\pi z_i) + 10]$	100	$[-5.12, 5.12]^n$	450.0
Noncontinuous Rastrigin	$f_4(\mathbf{x}) = \sum_{i=1}^{n}[y_i^2 - 10\cos(2\pi y_i) + 10]$ $y_i = \begin{cases} z_i & \lvert z_i \rvert < \frac{1}{2} \\ \frac{\text{round}(2z_i)}{2} & \lvert z_i \rvert \geq \frac{1}{2} \end{cases}$	100	$[-5.12, 5.12]^n$	180.0
Ackley	$f_5(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi z_i)\right) + 20 + e$	100	$[-32, 32]^n$	120.0
Griewank	$f_6(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{n} z_i^2 - \prod_{i=1}^{n}\cos(\frac{z_i}{\sqrt{i}}) + 1$	100	$[-600, 600]^n$	330.0
Generalized Penalized	$f_7(\mathbf{x}) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2$ $\times[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$ $+\sum_{i=1}^{n} u(z_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(z_i + 1)$ $u(z_i, a, k, m) = \begin{cases} k(z_i - a)^m & z_i > a, \\ 0 & -a < z_i < a \\ k(-z_i - a)^m & z_i < -a \end{cases}$	100	$[-50, 50]^n$	-330.0

Random Re-initialize Particles. Table 2 gives results of PSO with random re-initialization. A PSO with global star structure, initializing most particles randomly can promote diversity; particles have great ability of exploration. The middle and mean fitness value of every run has a reduction, which indicates that most fitness values are better than standard PSO.

Elitist Re-initialize Particles. Table 3 gives results of PSO with elitist re-initialization. A PSO with global star structure, re-initializing most particles can promote diversity; particles have great ability of exploration. The mean fitness value of every run also has a reduction at most times. Moreover, the ability of exploitation is increased than standard PSO, most fitness values, including best, middle, and mean fitness value are better than standard PSO. A PSO with local ring structure, which has elitist re-initialization strategy, can also obtain some improvement.

From the above results, we can see that an original PSO with local ring structure almost always has a better mean fitness value than PSO with global star structure. This illustrates that PSO with global star structure is easily deceived by local optima. Moreover, conclusion could be made that PSO with random or elitist re-initialization can promote PSO population diversity, i.e., increase ability of exploration, and not decrease ability of exploitation at the same time. Algorithms can get a better performance by utilizing this approach on multimodal problems.

Table 2. Representative results of PSO with random re-initialization. All algorithms have been run over 50 times, where "best", "middle", and"mean" indicate the best, middle, and mean of best fitness values for each run, respectively. Let $\beta = 500$, which means re-initialized part of particles after each 500 iterations, $\alpha \sim [0.05, 0.95]$ indicates that α fuzzy increased from 0.05 to 0.95 with step 0.05.

Result		Global Star Structure			Local Ring Structure		
		best	middle	mean	best	middle	mean
f_1	standard	287611.6	4252906.2	4553692.6	**-342.524**	-177.704	-150.219
	$\alpha \sim [0.05, 0.95]$	**13989.0**	**145398.5**	**170280.5**	-322.104	-188.030	**-169.959**
	$\alpha = 0.1$	132262.8	969897.7	1174106.2	-321.646	**-205.407**	-128.998
	$\alpha = 0.2$	195901.5	875352.4	1061923.2	-319.060	-180.141	-142.367
	$\alpha = 0.4$	117105.5	815643.1	855340.9	-310.040	-179.187	-52.594
f_4	standard	322.257	533.522	544.945	590.314	790.389	790.548
	$\alpha \sim [0.05, 0.95]$	**269.576**	**486.614**	**487.587**	**451.003**	**621.250**	**622.361**
	$\alpha = 0.1$	313.285	552.014	546.634	490.468	664.804	659.658
	$\alpha = 0.2$	285.430	557.045	545.824	520.750	654.771	659.538
	$\alpha = 0.4$	339.408	547.350	554.546	547.007	677.322	685.026
f_7	standard	36601631.0	890725077.1	914028295.8	-329.924	-327.990	-322.012
	$\alpha \sim [0.05, 0.95]$	**45810.66**	**2469089.3**	**5163181.2**	**-329.999**	-329.266	-311.412
	$\alpha = 0.1$	706383.80	77906145.5	85608026.9	**-329.999**	**-329.892**	**-329.812**
	$\alpha = 0.2$	4792310.46	60052595.2	82674776.8	-329.994	-329.540	-328.364
	$\alpha = 0.4$	238773.48	55449064.2	61673439.2	-329.991	-329.485	-329.435

Table 3. Representative results of PSO with elitist re-initialization. All algorithms have been run over 50 times, where "best", "middle", and"mean" indicate the best, middle, and mean of best fitness values for each run, respectively. Let $\beta = 500$, which means re-initialized part of particles after each 500 iterations, $\alpha \sim [0.05, 0.95]$ indicates that α fuzzy increased from 0.05 to 0.95 with step 0.05.

Result		Global Star Structure			Local Ring Structure		
		best	middle	mean	best	middle	mean
f_1	standard	287611.6	4252906.2	4553692.6	-342.524	-177.704	-150.219
	$\alpha \sim [0.05, 0.95]$	**23522.99**	1715351.9	1743334.3	306.371	-191.636	**-163.183**
	$\alpha = 0.1$	53275.75	**1092218.4**	**1326184.6**	**-348.058**	**-211.097**	-138.435
	$\alpha = 0.2$	102246.12	1472480.7	1680220.1	-340.859	-190.943	-90.192
	$\alpha = 0.4$	69310.34	1627393.6	1529647.2	-296.670	-176.790	-87.723
f_4	standard	322.257	533.522	544.945	590.314	790.389	790.548
	$\alpha \sim [0.05, 0.95]$	374.757	570.658	579.559	559.809	760.007	755.820
	$\alpha = 0.1$	371.050	564.467	579.968	538.227	**707.433**	**710.502**
	$\alpha = 0.2$	**314.637**	**501.197**	**527.120**	**534.501**	746.500	749.459
	$\alpha = 0.4$	352.850	532.293	533.687	579.000	773.282	764.739
f_7	standard	36601631.0	890725077	914028295	-329.924	-327.990	-322.012
	$\alpha \sim [0.05, 0.95]$	1179304.9	149747096	160016318	-329.889	-328.765	-328.707
	$\alpha = 0.1$	1213988.7	102300029	121051169	-329.998	**-329.784**	289.698
	$\alpha = 0.2$	1393266.07	**94717037**	**102467785**	-329.998	-329.442	**-329.251**
	$\alpha = 0.4$	**587299.33**	107998150	134572199	**-329.999**	-329.002	-328.911

5 Diversity Analysis and Discussion

Compared with other evolutionary algorithm, e.g., Genetic Algorithm, PSO has more search information, not only the solution (position), but also the velocity and cognitive. More information can be utilized to lead to a fast convergence; however, it also easily to be trapped to "local optima." Many approaches have been introduced based on the idea that prevents particles clustering too tightly in a region of the search space to achieve great possibility to "jump out" of local optima. However, these methods did not incorporate an effective way to measure the exploration/exploitation of particles.

Figure 1 displays the definitions of population diversities for variants of PSO. Firstly, the standard PSO: Fig.1 (a) and (b) display the population diversities of function f_1 and f_4. Secondly, PSO with random re-initialization: (c) and (d) display the diversities of function f_7 and f_1. The last is PSO with elitist re-initialization: (e) and (f) display the diversities of f_4 and f_9, respectively. Fig. 1 (a), (c), and (e) are for PSOs with global star structure, and others are PSO with local ring structure.

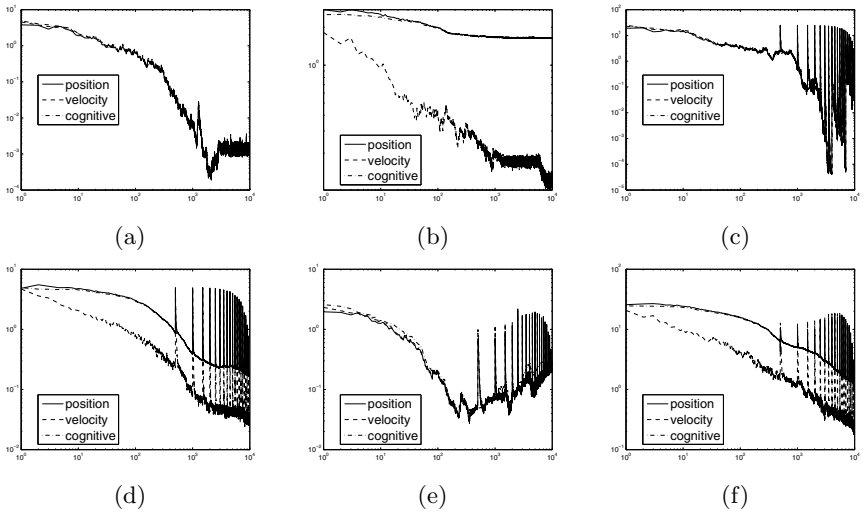


Fig. 1. Definitions of PSO population diversities. Original PSO: (a) f_1 global star structure, (b) f_4 local ring structure; PSO with random re-initialization: (c) f_7 global star structure, (d) f_1 local ring structure; PSO with elitist re-initialization: (e) f_4 global star structure, (f) f_7 local ring structure.

Figure 2 displays the comparison of population diversities for variants of PSO. Firstly, the PSO with global star structure: Fig.2 (a), (b) and (c) display function f_1 position diversity, f_4 velocity diversity, and f_7 cognitive diversity, respectively. Secondly, the PSO with local ring structure: (d), (e), and (f) display function f_1 velocity diversity, f_4 cognitive diversity, and f_7 position diversity, respectively.

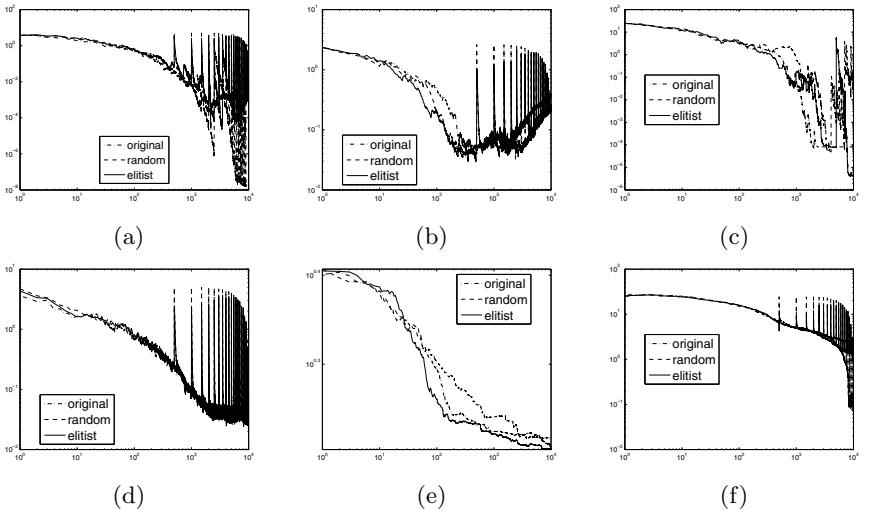


Fig. 2. Comparison of PSO population diversities. PSO with global star structure: (a) f_1 position, (b) f_4 velocity, (c) f_7 cognitive; PSO with local ring structure: (d) f_1 velocity, (e) f_4 cognitive, (f) f_7 position.

By looking at the shapes of the curves in all figures, it is easy to see that PSO with global star structure have more vibration than local ring structure. This is due to search information sharing in whole swarm, if a particle find a good solution, other particles will be influenced immediately. From the figures, it is also clear that PSO with random or elitist re-initialization can effectively increase diversity; hence, the PSO with re-initialization has more ability to "jump out" of local optima. Population diversities in PSO with re-initialization are promoted to avoid particles clustering too tightly in a region, and the ability of exploitation are kept to find "good enough" solution.

6 Conclusion

Low diversity, which particles clustering too tight, is often regarded as the main cause of premature convergence. This paper proposed two mechanisms to promote diversity in particle swarm optimization. PSO with random or elitist re-initialization can effectively increase population diversity, i.e., increase the ability of exploration, and at the same time, it can also slightly increase the ability of exploitation. To solve multimodal problem, great exploration ability means that algorithm has great possibility to "jump out" of local optima.

By examining the simulation results, it is clear that re-initialization has a definite impact on performance of PSO algorithm. PSO with elitist re-initialization, which increases the ability of exploration and keeps ability of exploitation at a same time, can achieve better results on performance. It is still imperative

to verify the conclusions found in this study in different problems. Parameters tuning for different problems are also needed to be researched.

The idea of diversity promoting can also be applied to other population-based algorithms, e.g., genetic algorithm. Population-based algorithms have the same concepts of population solutions. Through the population diversity measurement, useful information of search in exploration or exploitation state can be obtained. Increasing the ability of exploration, and keeping the ability of exploitation are beneficial for algorithm to "jump out" of local optima, especially when the problem to be solved is a computationally expensive problem.

References

1. Blackwell, T.M., Bentley, P.: Don't push me! collision-avoiding swarms. In: Proceedings of The Fourth Congress on Evolutionary Computation (CEC 2002), pp. 1691–1696 (May 2002)
2. Bratton, D., Kennedy, J.: Defining a standard for particle swarm optimization. In: Proceedings of the 2007 IEEE Swarm Intelligence Symposium, pp. 120–127 (2007)
3. Cheng, S., Shi, Y.: Diversity control in particle swarm optimization. In: Proceedings of the 2011 IEEE Swarm Intelligence Symposium, pp. 110–118 (April 2011)
4. Cheng, S., Shi, Y.: Normalized Population Diversity in Particle Swarm Optimization. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) ICSI 2011, Part I. LNCS, vol. 6728, pp. 38–45. Springer, Heidelberg (2011)
5. Clerc, M.: The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1951–1957 (July 1999)
6. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Processings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43 (1995)
7. Eberhart, R., Shi, Y.: Particle swarm optimization: Developments, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation, pp. 81–86 (2001)
8. Eberhart, R., Shi, Y.: Computational Intelligence: Concepts to Implementations. Morgan Kaufmann Publisher (2007)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Processings of IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
10. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publisher (2001)
11. Mendes, R., Kennedy, J., Neves, J.: The fully informed particle swarm: Simpler, maybe better. IEEE Transactions on Evolutionary Computation 8(3), 204–210 (2004)
12. Shi, Y., Eberhart, R.: Population diversity of particle swarms. In: Proceedings of the 2008 Congress on Evolutionary Computation, pp. 1063–1067 (2008)
13. Shi, Y., Eberhart, R.: Monitoring of particle swarm optimization. Frontiers of Computer Science 3(1), 31–37 (2009)
14. Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)
15. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation 3(2), 82–102 (1999)

Analysis of Feature Weighting Methods Based on Feature Ranking Methods for Classification

Norbert Jankowski and Krzysztof Usowicz

Department of Informatics, Nicolaus Copernicus University, Toruń, Poland

Abstract. We propose and analyze new fast feature weighting algorithms based on different types of feature ranking. Feature weighting may be much faster than feature selection because there is no need to find cut-threshold in the raking. Presented weighting schemes may be combined with several distance based classifiers like SVM, kNN or RBF network (and not only). Results shows that such method can be successfully used with classifiers.

Keywords: Feature weighting, feature selection, computational intelligence.

1 Introduction

Data used in classification problems consists of instances which typically are described by features (sometimes called attributes). The feature relevance (or irrelevance) differs between data benchmarks. Sometimes the relevance depends even on the classifier model, not only on data. Also the magnitude of feature may provide stronger or weaker influence on the usage of a given metric. What's more the values of feature may be represented in different units (keeping theoretically the same information) what may provide another source of problems (for example milligrams, kilograms, erythrocytes) for classifier learning process. This shows that feature selection must not be enough to solve a hidden problem. Obligatory usage of data standardization also must not be equivalent to the best way which can be done at all. It may happen that subset of features are for example counters of word frequencies. Then in case of normal data standardization will loose (almost) completely the information which was in a subset of features.

This is why we propose and investigate several methods of automated weighting of features instead of feature selection. Additional advantage of feature weighting over feature selection is that in case of feature selection there is not only the problem of choosing the ranking method but also of choosing the cut-threshold which must be validated what generates computational costs which are not in case of feature weighting. But not all feature weighting algorithms are really fast. The feature weightings which are wrappers (so adjust weights and validate in a long loop) [21,18,1,19,17] are rather slow (even slower than feature selection), however may be accurate. This provided us to propose several feature weighting methods based on feature ranking methods. Previously rankings were used to build feature weighting in [9] were values of mutual information were used directly as weights and in [24] used χ^2 distribution values for weighting. In this article we also present selection of appropriate weighting schemes which are used on values of rankings.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 238–247, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Below section presents chosen feature ranking methods which will be combined with designed weighting schemes that are described in the next section (3). Testing methodology and results of analysis of weighting methods are presented in section 4.

2 Selection of Rankings

The feature ranking selection is composed of methods which computation costs are relatively small. The computation costs of ranking should never exceed the computation costs of training and testing of final classifier (the kNN, SVM or another one) on average data stream. To make the tests more trustful we have selected ranking methods of different types as in [7]: based on correlation, based on information theory, based on decision trees and based on distance between probability distributions.

Some of the ranking methods are supervised and some are not. However all of them shown here are supervised. Computation of ranking values for features may be independent or dependent. What means that computation of next rank value may (but must not) depend on previously computed ranking values. For example Pearson correlation coefficient is independent while ranking based on decision trees or Battiti ranking are dependant.

Feature ranking may assign high values for relevant features and small for irrelevant ones or vice versa. First type will be called *positive feature ranking* and second *negative feature ranking*. Depending on this type the method of weighting will change its tactic.

For further descriptions assume that the data is represented by a matrix X which has m rows (the instances or vectors) and n columns called features. Let the $\mathbf{x}$ mean a single instance, $\mathbf{x}_i$ being i-th instance of X. And let's X_j means the j-th feature of X. In addition to X we have vector $\mathbf{c}$ of class labels.

Below we describe shortly selected ranking methods.

Pearson correlation coefficient ranking (CC): The Pearson's correlation coefficient:

$$CC(X_j, c) = \left[\sum_{i=1}^{m} (x_i^j - \bar{X}^j)(c_i - \bar{c}) \right] \Big/ (\sigma_{X^j} \cdot \sigma_c) \tag{1}$$

is really useful as feature selection [14,12]. $\bar{X}^j$ and σ_{X^j} means average value and standard deviation of j-th feature (and the same for vector $\mathbf{c}$ of class labels). Indeed the ranking values are absolute values of CC:

$$J_{CC}(X_j) = |CC(X^j, c)| \tag{2}$$

because correlation equal to -1 is indeed as informative as value 1. This ranking is simple to implement and its complexity is low $O(mn)$. However some difficulties arise when used for nominal features (with more then 2 values).

Fisher coefficient: Next ranking is based on the idea of Fisher linear discriminant and is represented as coefficient:

$$J_{FSC}(X_j) = \left[\bar{X}^{j,1} - \bar{X}^{j,2} \right] / \left[\sigma_{X^{j,1}} + \sigma_{X^{j,2}} \right], \tag{3}$$

where indices $j,1$ and $j,2$ mean that average (or standard deviation) is defined for j-th feature but only for either vectors of first or second class respectively. Performance

of feature selection using Fisher coefficient was studied in [11]. This criterion may be simply extended to multiclass problems.

χ^2 coefficient: The last ranking in the group of correlation based method is the χ^2 coefficient:

$$J_{\chi^2}(X_j) = \sum_{i=1}^{m}\sum_{k=1}^{l} \frac{\left[p(X_j = x_i^j, C = c_k) - p(X_j = x_i^j)p(C = c_k)\right]^2}{p(X_j = x_i^j)p(C = c_k)}. \tag{4}$$

Using this method in context of feature selection was discussed in [8]. This method was also proposed for feature weighting with the kNN classifier in [24].

2.1 Information Theory Based Feature Rankings

Mutual Information Ranking (MI): Shannon [23] described the concept of entropy and mutual information. Now the concept of entropy and mutual information is widely used in several domains. The entropy in context of feature may be defined by:

$$H(X_j) = -\sum_{i=1}^{m} p(X_j = x_i^j)\log_2 p(X_j = x_i^j) \tag{5}$$

and in similar way for class vector: $H(\mathbf{c}) = -\sum_{i=1}^{m} p(C = c_i)\log_2 p(C = c_i)$.

The *mutual information (MI)* may be used as a base of feature ranking:

$$J_{MI}(X_j) = I(X_j, \mathbf{c}) = H(X_j) + H(\mathbf{c}) - H(X_j, \mathbf{c}), \tag{6}$$

where $H(X_j, \mathbf{c})$ is joint entropy. Mutual information was investigated as ranking method several times [3, 14, 8, 13, 16]. The MI was also used for feature weighting in [9].

Asymmetric Dependency Coefficient (ADC) is defined as mutual information normalized by entropy of classes:

$$J_{ADC}(X_j) = I(X_j, \mathbf{c})/H(\mathbf{c}). \tag{7}$$

These and next criterions which base on MI were investigated in context of feature ranking in [8, 7].

Normalized Information Gain (US) proposed in [22] is defined by the MI normalized by the entropy of feature:

$$J_{ADC}(X_j) = I(X_j, \mathbf{c})/H(X_j). \tag{8}$$

Normalized Information Gain (UH) is the third possibility of normalizing, this time by the joint entropy of feature and class:

$$J_{UH}(X_j) = I(X_j, \mathbf{c})/H(X_j, \mathbf{c}). \tag{9}$$

Symmetrical Uncertainty Coefficient (SUC): This time the MI is normalized by the sum of entropies [15]:

$$J_{SUC}(X_j) = I(X_j, \mathbf{c})/(H(X_j, \mathbf{c}) + H(\mathbf{c})). \tag{10}$$

It can be simply seen that the normalization is like weight modification factor which has influence in the order of ranking and in pre-weights for further weighting calculation.

Except the DML all above MI-based coefficients compose positive rankings.

2.2 Decision Tree Rankings

Decision trees may be used in a few ways for feature selection or ranking building. The simplest way of feature selection is to select features which were used to build the given decision tree to play the role of the classifier. But it is possible to compose not only a binary ranking, the criterion used for the tree node selection can be used to build the ranking.

The selected decision trees are: CART [4], C4.5 [20] and SSV [10]. Each of those decision trees uses its own split criterion, for example CART use the GINI or SSV use the separability split value. For using SSV in feature selection please see [11].

The feature ranking is constructed basing on the nodes of decision tree and features used to build this tree. Each node is assigned to a split point on a given feature which has appropriate value of the split criterion. These values will be used to compute ranking according to:

$$J(X_j) = \sum_{n \in Q_j} split(n), \tag{11}$$

where Q_j is a set of nodes which split point uses feature j, and $split(n)$ is the value of given split criterion for the node n (depend on tree type). Note that features not used in tree are not in the ranking and in consequence will have weight 0.

2.3 Feature Rankings Based on Probability Distribution Distance

Kolmogorov distribution distance (KOL) based ranking was presented in [7]:

$$J_{KOL}(X_j) = \sum_{i=1}^{m} \sum_{k=1}^{l} \left| p(X_j = x_i^j, C = c_k) - p(X_j = x_i^j) p(C = c_k) \right| \tag{12}$$

Jeffreys-Matusita Distance (JM) is defined similarly to the above ranking:

$$J_{JM}(X_j) = \sum_{i=1}^{m} \sum_{k=1}^{l} \left| \sqrt{p(X_j = x_i^j, C = c_k)} - \sqrt{p(X_j = x_i^j) p(C = c_k)} \right|^2 \tag{13}$$

MIFS ranking. Battiti [3] proposed another ranking which bases on MI. In general it is defined by:

$$J_{MIFS}(X_j|S) = I((X_j, \mathbf{c})|S) = I(X_j, \mathbf{c}) - \beta \cdot \sum_{s \in S} I(X_j, X_s). \tag{14}$$

This ranking is computed iteratively basing on previously established ranking values. First, as the best feature, the j-th feature which maximizes $I(X_J, \mathbf{c})$ (for empty S) is chosen. Next the set S consists of index of first feature. Now the second winner feature has to maximize right side of Eq. 14 with the sum over non-empty S. Next ranking values are computer in the same way.

To eliminate the parameter β Huang et. al [16] proposed a changed version of Eq.14:

$$J_{SMI}(X_j|S) = I(X_j,\mathbf{c}) - \sum_{s\in S}\left[\frac{I(X_j,X_s)}{H(X_s)} - \frac{1}{2}\sum_{s'\in S, s'\neq s}\frac{I(X_j,X_{s'})}{H(X_{s'})}\cdot\frac{I(X_{s'},X_s)}{H(X_s)}\right]\cdot I(X_s,\mathbf{c}). \tag{15}$$

The computation of J_{SMI} is done in the same way as J_{MIFS}. Please note that computation of J_{MIFS} and J_{SMI} is more complex then computation of previously presented rankings that base on MI.

Fusion Ranking (FUS). Resulting feature rankings may be combined to another ranking in *fusion* [25]. In experiments we combine six rankings (NMF, NRF, NLF, NSF, MDF, SRW[1]) as their sum. However an different operator may replace the sum (median, max, min). Before calculation of fusion ranking each ranking used in fusion has to be normalized.

3 Methods of Feature Weighting for Ranking Vectors

Direct use of ranking values to feature weighting is sometimes even impossible because we have positive and negative rankings. However in case of some rankings it is possible [9,6,5]. Also the character of magnitude of ranking values may change significantly between kinds of ranking methods[2]. This is why we decided to check performance of a few weighting schemes while using every single one with each feature ranking method.

Below we propose methods which work in one of two types of weighting schemes: first use the ranking values to construct the weight vector while second scheme uses the order of features to compose weight vector.

Let's assume that we have to weight vector of feature ranking $\mathbf{J} = [j_1,\ldots,J_n]$. Additionally define $J_{min} = \min_{i=1,\ldots,n} J_i$ and $J_{max} = \max_{i=1,\ldots,n} J_i$.

Normalized Max Filter (NMF) is Defined by

$$\mathscr{W}_{NMF}(J) = \begin{cases} |J|/J_{max} & \text{for} \quad J+ \\ [J_{max}+J_{min}-|J|]/J_{max} & \text{for} \quad J- \end{cases}, \tag{16}$$

where J is ranking element of $\mathbf{J}$. $J+$ means that the feature ranking is positive and $J-$ means negative ranking. After such transformation the weights lie in $[J_{min},J_{max},1]$.

Normalizing Range Filter (NRF) is a bit similar to previous weighting function:

$$\mathscr{W}_{NRF}(J) = \begin{cases} (|J|+J_{min})/(J_{max}+J_{min}) & \text{for} \quad J+ \\ (J_{max}+2J_{min}-|J|)/(J_{max}+J_{min}) & \text{for} \quad J- \end{cases}. \tag{17}$$

In such case weights will lie in $[2J_{min}/(J_{max}+J_{min}),1]$.

Normalizing Linear Filter (NLF) is another a linear transformation defined by:

$$\mathscr{W}_{NLF}(J) = \begin{cases} \frac{[1-\varepsilon]J+[\varepsilon-1]J_{max}}{J_{max}-J_{min}} & \text{for} \quad J+ \\ \frac{[\varepsilon-1]J+[1-\varepsilon]J_{max}}{J_{max}-J_{min}} & \text{for} \quad J- \end{cases}, \tag{18}$$

[1] See Eq. 21.

[2] Compare sequence $1,2,3,4$ with $11,12,13,14$ further influence in metric is significantly different

where $\varepsilon = -(\varepsilon_{max} - \varepsilon_{min})v^p + \varepsilon_{max}$ depends on feature. Parameters has typically values: $\varepsilon_{min} = 0.1$ and $\varepsilon_{max} = 0.9$, and p may be 0.25 or 0.5. And $v = \sigma_{\mathbf{J}}/\bar{\mathbf{J}}$ is a variability index.

Normalizing Sigmoid Filter (NSF) is a nonlinear transformation of ranking values:

$$W_{NSF}(J) = \left[1 + e^{-[\mathscr{W}(J)-0.5]\log((1-\varepsilon')/\varepsilon')^2}\right]^{-1} + \varepsilon' \tag{19}$$

where $\varepsilon' = \varepsilon/2$. This weighting function increases the strength of strong features and decreases weak features.

Monotonically Decreasing Function (MDF) defines weights basing on the order of the features, not on the ranking values:

$$W_{MDF}(j) = e^{\log\varepsilon\cdot[(j-1)/(n-1)]^{\log_{(n_s-1)/(n-1)}\log_\varepsilon\tau}} \tag{20}$$

where j is the position of the given feature in order. τ may be 0.5. Roughly it means the n_s/n fraction of features will have weights not greater than tau.

Sequential Ranking Weighting (SRW) is a simple threshold weighting via feature order:

$$W_{SRW}(j) = [n+1-j]/n, \tag{21}$$

where j is again the position in the order.

4 Testing Methodology and Results Analysis

The test were done on several benchmarks from UCI machine learning repository [2]: appendicitis, Australian credit approval, balance scale, Wisconsin breast cancer, car evaluation, churn, flags, glass identification, heart disease, congressional voting records, ionosphere, iris flowers, sonar, thyroid disease, Telugu vowel, wine.

Each single test configuration of a weighting scheme and a ranking method was tested using 10 times repeater 10 fold cross-validation (CV). Only the accuracies from testing parts of CV were used in further test processing. In place of presenting averaged accuracies over several benchmarks the paired t-tests were used to count how many times had the given test configuration won, defeated or drawn. t-test is used to compare efficiency of a classifiers without weighting and with weighting (a selected ranking method plus selected weighting scheme). For example efficiency of 1NNE classifier (one nearest neighbour with Euclidean metric) is compared to 1NNE with weighting by CC ranking and NMF weighting scheme. And this is repeated for each combination of rankings and weighting schemes. CV tests of different configurations were using the same random seed to make the test more trustful (it enables the use of paired t-test).

Table 1 presents results averaged for different configurations of k nearest neighbors kNN and SVM: 1NNE, 5NNE, AutoNNE, SVME, AutoSVME, 1NNM, 5NNM, AutoNNM, SVMM, AutoSVMM. Were suffix ‘E’ or ‘M’ means Euclidean or Manhattan respectively. Prefix ‘auto’ means that kNN chose the ‘k’ automatically or SVM chose the ‘C’ and spread of Gaussian function automatically.

Tables 1(a)–(c) presents counts of winnings, defeats and draws. Is can be seen that the best choice of ranking method were US, UH and SUC while the best weighting schemes

Table 1. Cumulative counts over feature ranking methods and feature weighting schemes (averaged over kNN's and SVM's configurations)

(a) Cumulative count of winnings

40.53%	NMF	NRF	NLF	**NSF**	**MDF**	SRW	*Sum*
CC	55	59	53	59	55	52	*333*
FSC	61	58	63	72	64	64	*382*
CHI	67	66	**78**	73	66	72	*422*
MI	66	70	74	76	67	73	*426*
ADC	66	70	74	76	67	73	*426*
US	75	71	69	**79**	76	71	***441***
UH	73	71	**77**	**77**	71	71	***440***
DML	58	54	59	58	58	49	*336*
SUC	71	70	**79**	**77**	70	71	***438***
CRT	46	46	44	43	66	66	*311*
C45	46	48	41	42	64	63	*304*
SSV	50	48	48	48	63	62	*319*
KOL	62	68	66	64	68	67	*395*
JM	72	70	74	76	69	69	*430*
SMI	57	59	64	68	71	69	*388*
FUS	**78**	73	76	74	68	65	*434*
Sum	1003	1001	1039	**1062**	**1063**	1057	*6225*

(b) Cumulative count of defeats

36.33%	NMF	NRF	NLF	**NSF**	**MDF**	SRW	*Sum*
CC	67	61	61	53	66	76	*384*
FSC	68	63	61	56	65	66	*379*
CHI	55	55	59	51	52	58	*330*
MI	53	52	59	51	52	57	*324*
ADC	53	52	59	51	52	57	*324*
US	55	47	58	45	50	60	*315*
UH	53	53	54	51	51	57	*319*
DML	41	36	56	46	58	66	*303*
SUC	52	47	57	49	51	57	*313*
CRT	92	83	94	89	52	59	*469*
C45	87	83	86	83	60	63	*462*
SSV	79	72	82	76	55	60	*424*
KOL	50	42	52	39	46	51	***280***
JM	55	52	56	50	55	56	*324*
SMI	69	68	63	56	55	56	*367*
FUS	35	33	51	36	50	58	***263***
Sum	964	899	1008	**882**	**870**	957	*5580*

(c) Cumulative count of draws

23.14%	NMF	NRF	NLF	NSF	MDF	SRW	*Sum*
CC	38	40	46	48	39	32	*243*
FSC	31	39	36	32	31	30	*199*
CHI	38	39	23	36	42	30	*208*
MI	41	38	27	33	41	30	*210*
ADC	41	38	27	33	41	30	*210*
US	30	42	33	36	34	29	*204*
UH	34	36	29	32	38	32	*201*
DML	61	70	45	56	44	45	*321*
SUC	37	43	24	34	39	32	*209*
CRT	22	31	22	28	42	35	*180*
C45	27	29	33	35	36	34	*194*
SSV	31	40	30	36	42	38	*217*
KOL	48	50	42	57	46	42	*285*
JM	33	38	30	34	36	35	*206*
SMI	34	33	33	36	34	35	*205*
FUS	47	54	33	50	42	37	*263*
Sum	593	660	513	616	627	546	*3555*

(d) Cumulative counts per classifier configuration

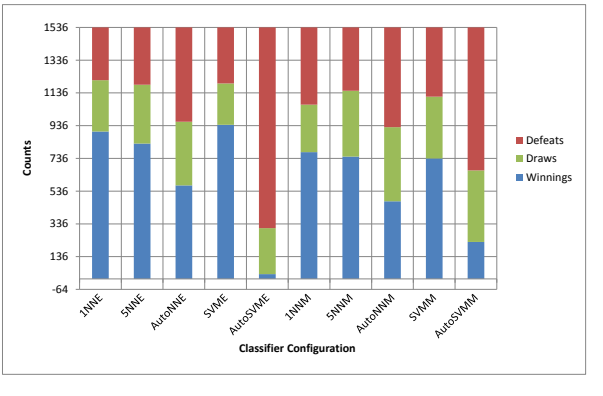


Table 2. Cumulative counts over feature ranking methods and feature weighting schemes for SVM classifier

(a) Cumulative count of winnings

61,26%	NMF	NRF	NLF	NSF	MDF	SRW	*Sum*
CC	9	9	9	11	11	11	***60***
FSC	9	7	8	9	10	10	*53*
CHI	10	10	11	11	9	10	***61***
MI	11	11	11	11	9	9	***62***
ADC	11	11	11	11	9	9	***62***
US	11	12	10	10	10	9	***62***
UH	10	11	10	11	9	8	*59*
DML	8	9	9	9	9	8	*52*
SUC	11	12	10	10	9	8	***60***
CRT	10	9	10	9	10	9	*57*
C45	10	9	9	9	9	8	*54*
SSV	10	8	9	10	9	10	*56*
KOL	9	9	10	11	10	9	*58*
JM	11	11	11	12	10	11	***66***
SMI	10	10	9	10	10	11	***60***
FUS	10	11	10	10	10	8	*59*
Sum	**160**	**159**	157	**164**	153	148	*941*

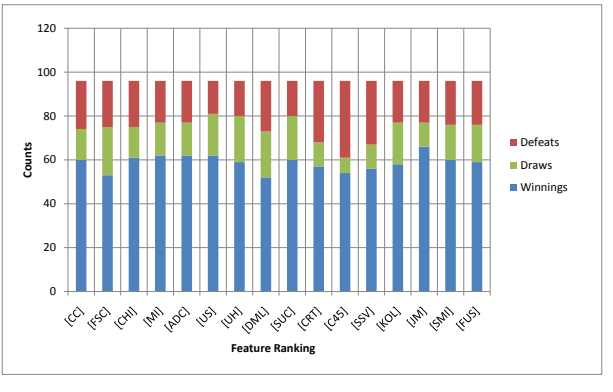
(b) Cumulative count of defeats

22.27%	NMF	NRF	NLF	NSF	MDF	SRW	*Sum*
CC	4	3	4	3	4	4	*22*
FSC	3	3	4	3	4	4	*21*
CHI	3	3	4	3	4	4	*21*
MI	2	2	4	3	4	4	*19*
ADC	2	2	4	3	4	4	*19*
US	2	1	3	2	3	4	***15***
UH	2	2	3	3	3	3	***16***
DML	3	3	5	3	4	5	*23*
SUC	2	2	3	3	3	3	***16***
CRT	4	5	4	5	5	5	*28*
C45	6	6	6	6	6	5	*35*
SSV	5	5	5	5	4	5	*29*
KOL	2	3	4	3	4	3	*19*
JM	2	2	4	3	4	4	*19*
SMI	4	4	3	2	4	3	*20*
FUS	3	3	4	3	4	3	*20*
Sum	**49**	**49**	64	**53**	64	63	*342*

(c) Cumulative count of draws

16.47%	NMF	NRF	NLF	NSF	MDF	SRW	*Sum*
CC	3	4	3	2	1	1	*14*
FSC	4	6	4	4	2	2	***22***
CHI	3	3	1	2	3	2	*14*
MI	3	3	1	2	3	3	*15*
ADC	3	3	1	2	3	3	*15*
US	3	3	3	4	3	3	***19***
UH	4	3	3	2	4	5	***21***
DML	5	4	2	4	3	3	***21***
SUC	3	2	3	3	4	5	*20*
CRT	2	2	2	2	1	2	*11*
C45	0	1	1	1	1	3	*7*
SSV	1	3	2	1	3	1	*11*
KOL	5	4	2	2	2	4	*19*
JM	3	3	1	1	2	1	*11*
SMI	2	2	4	4	2	2	*16*
FUS	3	2	2	3	2	5	*17*
Sum	**47**	**48**	35	39	39	**45**	*253*

(d) Cumulative counts per classifier configuration



were NSF and MDF in average. Smaller number of defeats were obtained for KOL and FUS rankings and for NSF and MDF weighting schemes. Over all best configuration is combination of US ranking with NSF weighting scheme. The worst performance characterize feature rankings based on decision trees. Note that the weighting with a classifier must not be used obligatory. With a help of CV validation it may be simply verified whether the using of feature weighting method for given problem (data) can be recommended or not. Table 1(d) presents counts of winnings, defeats and draws per classification configuration. The highest number of winnings were obtained for SVME, 1NNE, 5NNE. The weighting turned out useless for AutoSVM[E|M]. This means that weighting does not help in case of internally optimized configurations of SVM. But note that optimization of SVM is much more costly (around 100 times—costs of grid validation) than SVM with feature weighting!

Tables 2(a)–(d) describe results for SVME classifier used with all combinations of weighting as before. Weighting for SVM is very effective even with different rankings (JM, MI, ADC, US,CHI, SUC or SMI) and with weighting schemes: NSF, NMF, NRF.

5 Summary

Presented feature weighting methods are fast and accurate. In most cases performance of the classifier may be increased without significant growth of computational costs. The best weighting methods are not difficult to implement. Some combinations of ranking and weighting schemes are often better than other, for example combination of normalized information gain (US) and NSF. Presented feature weighting methods may compete with slower feature selection or adjustment methods of classifier meta-parameters (AutokNN or AutoSVM which needs slow parameters tuning). By simple validation we may decide whether to weight or not to weight features before using the chosen classifier for given data (problem) keeping the final decision model more accurate.

References

1. Aha, D.W., Goldstone, R.: Concept learning and flexible weighting. In: Proceedings of the 14th Annual Conference of the Cognitive Science Society, pp. 534–539 (1992)
2. Asuncion, A., Newman, D.: UCI machine learning repository (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
3. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks 5(4), 537–550 (1994)
4. Breiman, L., Friedman, J.H., Olshen, A., Stone, C.J.: Classification and regression trees. Wadsworth, Belmont (1984)
5. Creecy, R.H., Masand, B.M., Smith, S.J., Waltz, D.L.: Trading mips and memory for knowledge engineering. Communications of the ACM 35, 48–64 (1992)
6. Daelemans, W., van den Bosch, A.: Generalization performance of backpropagation learning on a syllabification task. In: Proceedings of TWLT3: Connectionism and Natural Language Processing, pp. 27–37 (1992)
7. Duch, W.: Filter methods. In: Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds.) Feature Extraction, Foundations and Applications. Studies in fuzziness and soft computing, pp. 89–117. Springer, Heidelberg (2006)

8. Duch, W., Biesiada, T.W.J., Blachnik, M.: Comparison of feature ranking methods based on information entropy. In: Proceedings of International Joint Conference on Neural Networks, pp. 1415–1419. IEEE Press (2004)
9. Wettschereck, D., Aha, D., Mohri, T.: A review of empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artificial Intelligence Review Journal 11, 273–314 (1997)
10. Grąbczewski, K., Duch, W.: The separability of split value criterion. In: Rutkowski, L., Tadeusiewicz, R. (eds.) Neural Networks and Soft Computing, Zakopane, Poland, pp. 202–208 (June 2000)
11. Grąbczewski, K., Jankowski, N.: Feature selection with decision tree criterion. In: Nedjah, N., Mourelle, L., Vellasco, M., Abraham, A., Köppen, M. (eds.) Fifth International conference on Hybrid Intelligent Systems, pp. 212–217. IEEE Computer Society, Brasil (2005)
12. Grąbczewski, K., Jankowski, N.: Mining for complex models comprising feature selection and classification. In: Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds.) Feature Extraction, Foundations and Applications. Studies in fuzziness and soft computing, pp. 473–489. Springer, Heidelberg (2006)
13. Guyon, I.: Practical feature selection: from correlation to causality. 955 Creston Road, Berkeley, CA 94708, USA (2008), http://eprints.pascal-network.org/archive/00004038/01/PracticalFS.pdf
14. Guyon, I., Elisseef, A.: An introduction to variable and feature selection. Journal of Machine Learning Research, 1157–1182 (2003)
15. Hall, M.A.: Correlation-based feature subset selection for machine learning. Ph.D. thesis, Department of Computer Science, University of Waikato, Waikato, New Zealand (1999)
16. Huang, J.J., Cai, Y.Z., Xu, X.M.: A parameterless feature ranking algorithm based on MI. Neurocomputing 71, 1656–1668 (2007)
17. Jankowski, N.: Discrete quasi-gradient features weighting algorithm. In: Rutkowski, L., Kacprzyk, J. (eds.) Neural Networks and Soft Computing. Advances in Soft Computing, pp. 194–199. Springer, Zakopane (2002)
18. Kelly, J.D., Davis, L.: A hybrid genetic algorithm for classification. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence, pp. 645–650 (1991)
19. Kira, K., Rendell, L.A.: The feature selection problem: Traditional methods and a new algorithm. In: Proceedings of the 10th International Joint Conference on Artificial Intelligence, pp. 129–134 (1992)
20. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann, San Mateo (1993)
21. Salzberg, S.L.: A nearest hyperrectangle learning method. Machine Learning Journal 6(3), 251–276 (1991)
22. Setiono, R., Liu, H.: Improving backpropagation learning with feature selection. Applied Intelligence 6, 129–139 (1996)
23. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal 27, 379–423, 623–656 (1948)
24. Vivencio, D.P., Hruschka Jr., E.R., Nicoletti, M., Santos, E., Galvao, S.: Feature-weighted k-nearest neigbor classifier. In: Proceedings of IEEE Symposium on Foundations of Computational Intelligence (2007)
25. Yan, W.: Fusion in multi-criterion feature ranking. In: 10th International Conference on Information Fusion, pp. 1–6 (2007)

Simultaneous Learning of Instantaneous and Time-Delayed Genetic Interactions Using Novel Information Theoretic Scoring Technique

Nizamul Morshed, Madhu Chetty, and Nguyen Xuan Vinh

Monash University,
Australia
{nizamul.morshed,madhu.chetty,vinh.nguyen}@monash.edu

Abstract. Understanding gene interactions is a fundamental question in systems biology. Currently, modeling of gene regulations assumes that genes interact either instantaneously or with time delay. In this paper, we introduce a framework based on the Bayesian Network (BN) formalism that can represent both instantaneous and time-delayed interactions between genes simultaneously. Also, a novel scoring metric having firm mathematical underpinnings is then proposed that, unlike other recent methods, can score both interactions concurrently and takes into account the biological fact that multiple regulators may regulate a gene jointly, rather than in an isolated pair-wise manner. Further, a gene regulatory network inference method employing evolutionary search that makes use of the framework and the scoring metric is also presented. Experiments carried out using synthetic data as well as the well known *Saccharomyces cerevisiae* gene expression data show the effectiveness of our approach.

Keywords: Information theory, Bayesian network, Gene regulatory network.

1 Introduction

In any biological system, various genetic interactions occur amongst different genes concurrently. Some of these genes would interact almost instantaneously while interactions amongst some other genes could be time delayed. From biological perspective, instantaneous regulations represent the scenarios where the effect of a change in the expression level of a regulator gene is carried on to the regulated gene (almost) instantaneously. In these cases, the effect will be reflected almost immediately in the regulated gene's expression level[1]. On the other hand, in cases where regulatory interactions are time-delayed in nature, the effect may be seen on the regulated gene after some time.

Bayesian networks and its extension, dynamic Bayesian networks (DBN) have found significant applications in the modeling of genetic interactions [1,2]. To the

[1] The time-delay will always be greater than zero. However, if the delay is small enough so that the regulated gene is effected before the next data sample is taken, it can be considered as an instantaneous interaction.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 248–257, 2011.
© Springer-Verlag Berlin Heidelberg 2011

best of our knowledge, barring few exceptions (to be discussed in Section 2), all the currently existing gene regulatory network (GRN) reconstruction techniques that use time series data assume that the effect of changes in the expression level of a regulator gene is either instantaneous or maintains a d-th order Markov relation with its regulated gene (i.e., regulations occur between genes in two time slices, which can be at most d time steps apart, $d = 1, 2, \ldots$). In this paper, we introduce a framework (see Fig. 1) that captures both types of interactions. We also propose a novel scoring metric that takes into account the biological fact that multiple genes may regulate a single gene in a combined manner, rather than in an individual pair-wise manner. Finally, we present a GRN inference algorithm employing evolutionary search strategy that makes use of the framework and the scoring metric. The rest of the paper is organized as follows. In Section 2, we explain the framework that allows us to represent both instantaneous and time-delayed interactions simultaneously. This section also contains the related literature review and explains how these methods relate to our approach. Section 3 formalizes the proposed scoring metric and explains some of its theoretical properties. Section 4 describes the employed search strategy. Section 5 discusses the synthetic and real-life networks used for assessing our approach and also its comparison with other techniques. Section 6 provides concluding observations and remarks.

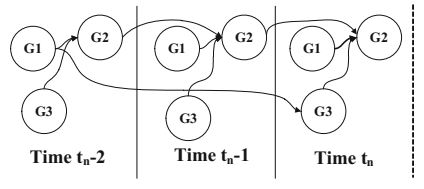


Fig. 1. Example of network structure with both instantaneous and time-delayed interactions

2 The Representational Framework

Let us model a gene network containing n genes (denoted by $X_1, X_2 \ldots, X_n$) with a corresponding microarray dataset having N time points. A DBN-based GRN reconstruction method would try to find associations between genes X_i and X_j by taking into consideration the data $x_{i1}, \ldots, x_{i(N-\delta)}$ and $x_{j(1+\delta)}, \ldots, x_{jN}$ or vice versa (small case letters mean data values in the microarray), where $1 \leq \delta \leq d$. This will effectively enable it to capture d-step time delayed interactions (at most). Conversely, a BN-based strategy would use the whole N time points and it will capture regulations that are effective instantaneously.

Now, to model both instantaneous and multiple step time-delayed interactions, we double the number of nodes as shown in Fig. 2. The zero entries in the figure denote no regulation. For the first n columns, the entries marked by 1 correspond to instantaneous regulations whereas for the last n columns non-zero entries denote the order of regulation.

Prior works on inter and intra-slice connections in dynamic probabilistic network formalism [3,4] have modelled a DBN using an initial network and a transition network employing the 1st-order Markov assumption, where the initial network exists only during the initial period of time and afterwards the dynamics is expressed using only the transition network. Realising that a d-th order DBN has variables replicated d times, a 1st-order DBN for this task[2] is therefore usually limited to, around 10 variables or a 2nd-order DBN can mostly deal with 6-7 variables [5]. Thus, prior works on DBNs either could not discover these two interactions simultaneously or were unable to fully exploit its potential restricting studies to simpler network configurations. However, since our proposed approach does not replicate variables, we can study any complex network configurations without limitations on the number of nodes. Zou et al. [2], while highlighting existence of both instantaneous and time-delayed interactions among genes while considering parent-child relationships of a particular order, did not account for the regulatory effects of other parents having different order. Our proposed method supports that multiples parents may regulate a child simultaneously, with different orders of regulation. Moreover, the limitation of detecting basic genetic interactions like $A \leftrightarrow B$ is also overcome with the proposed method.

Complications in the alignment of data samples can arise if the parents have different order of regulation with the child node. We elucidate this using an example, where we have already assessed the degree of interest (in terms of Mutual Information) in adding two parents (gene B and C, having third and first order regulations, respectively) to a gene under consideration, X. Now, we want to assess the degree of interest in adding gene A as a parent of X with a second order regulatory relationship (i.e., $MI(X, A^2|\{B^3, C^1\})$, where superscripts on the parent variables denote the order of regulation it has with the child node).

There are two possibilities: the first one corresponds to the scenario where the data is not periodic. In this case, we have to use $(N - \delta)$ samples where δ is the maximum order of regulation that the gene under consideration has, with its parent nodes (3 in this example). Fig. 3 shows how the alignment of the samples can be done for the current example. The $\surd$ symbol inside a cell denotes that this data sample will be used during MI computation, whereas empty cells denote that these data samples will not be considered. Similar alignments will need to be done for the other case, where the data is periodic (e.g., datasets of yeast compiled by [6] show such behavior [7]). However, we can use all the N data samples in this case.

Finally, the interpretation of the results obtained from an algorithm that uses this framework can be done in a straightforward manner. So, using this framework and the aligned data samples, if we construct a network where we observe, for example, arc $X_1 \rightarrow X_n^{'}$ having order δ, we conclude that the inter-slice arc between X_1 and X_n is inferred and X_1 regulates X_n with a δ-step time-delay. Similarly, if we find arc $X_2 \rightarrow X_n$, we say that the intra-slice arc between X_2 and X_n is inferred and a change in the expression level of X_2 will

[2] A tutorial can be found in
http://www.cs.ubc.ca/~murphyk/Software/BDAGL/dbnDemo_hus.htm

	X_1	X_2	...	X_n	X_1'	X_2'	...	X_n'
X_1	0	1	...	0	2	0	...	1
X_2	0	0	...	1	d	0	...	0
...	...	...	...	...	...	...	...	...
X_n	1	0	...	0	0	1	...	d

Fig. 2. Conceptual view of proposed approach

	1	2	3	4	...	N-3	N-2	N-1	N
A		√	√	√	...	√	√		
X				√	...	√	√	√	√
B	√	√	√	√	...	√			
C			√	√	...	√	√	√	

Fig. 3. Calculation of Mutual Information (MI)

almost immediately effect the expression level of X_n. The following 3 conditions must also be satisfied in any resulting network:

1. The network must be a directed acyclic graph.
2. The inter-slice arcs must go in the correct direction (no backward arc).
3. Interactions remain existent independent of time (Stationarity assumption).

3 Our Proposed Scoring Metric, CCIT

The proposed CCIT (Combined Conditional Independence Tests) score, when applied to a graph G containing n genes (denoted by $X_1, X_2 \ldots, X_n$), with a corresponding microarray dataset D, is shown in (1). The score relies on the decomposition property of MI and a theorem of Kullback [8].

$$S_{CCIT}(G{:}D) = \sum_{\substack{i=1 \\ Pa(X_i)\neq\phi}}^{n} \{\, 2N_{\delta_i}.MI(X_i, Pa(X_i)) - \sum_{k=0}^{\delta_i} (\max_{\sigma_i^k} \sum_{j=1}^{s_i^k} \chi_{\alpha, l_i \sigma_i^k(j)}) \,\} \tag{1}$$

Here s_i^k denotes the number of parents of gene X_i having a k step time-delayed regulation and δ_i is the maximum time-delay that gene X_i has with its parents. The parent set of gene X_i, $Pa(X_i)$ is the union of the parent sets of X_i having zero time-delay (denoted by $Pa^0(X_i)$), single-step time-delay ($Pa^1(X_i)$) and up to parents having the maximum time-delay (δ_i) and defined as follows:

$$Pa(X_i) = Pa^0(X_i) \cup Pa^1(X_i) \cdots \cup Pa^{\delta_i}(X_i) \tag{2}$$

The number of *effective* data points, N_{δ_i}, depends on whether the data can be considered to be showing periodic behavior or not (e.g., datasets compiled by [6] can be considered as showing periodic behavior [7]), and it is defined as follows:

$$N_{\delta_i} = \begin{cases} N & \text{if data is periodic} \\ N - \delta_i & \text{otherwise} \end{cases} \tag{3}$$

Finally, $\sigma_i^k = (\sigma_i^k(1), \ldots, \sigma_i^k(s_i^k))$ denote any permutation of the index set $(1, \ldots, s_i^k)$ of the variables $Pa^k(X_i)$ and $l_{i\sigma_i^k(j)}$, the degrees of freedom, is defined as follows:

$$l_{i\sigma_i^k(j)} = \begin{cases} (r_i - 1)(r_{\sigma_i^k(j)} - 1) \prod_{m=1}^{j-1} r_{\sigma_i^k(m)}, & \text{for } 2 \le j \le s_i^k \\ (r_i - 1)(r_{\sigma_i^k(1)} - 1), & \text{for } j = 1 \end{cases} \tag{4}$$

where r_p denotes the number of possible values that gene X_p can take (after discretization, if the data is continuous). If the number of possible values that the genes can take is not the same for all the genes, the quantity σ_i^k denotes the permutation of the parent set $Pa^k(X_i)$ where the first parent gene has the highest number of possible values, the second gene has the second highest number of possible values and so on.

The CCIT score is similar to those metrics which are based on maximizing a penalized version of the log-likelihood, such as BIC/MDL/MIT. However, unlike BIC/MDL, the penalty part in this case is local for each variable and its parents, and takes into account both the complexity and reliability of the structure. Also, both CCIT and MIT have the additional strength that the tests quantify the extent to which the genes are independent. Finally, unlike MIT [9], CCIT scores both intra and inter-slice interactions simultaneously, rather than considering these two types of interactions in an isolated manner, making it specially suitable for problems like reconstructing GRNs, where joint regulation is a common phenomenon.

3.1 Some Properties of CCIT Score

In this section we study several useful properties of the proposed scoring metric. The first among these is the decomposability property, which is especially useful for local search algorithms:

Proposition 1. *CCIT is a decomposable scoring metric.*

Proof. This result is evident as the scoring function is, by definition, a sum of local scores. □

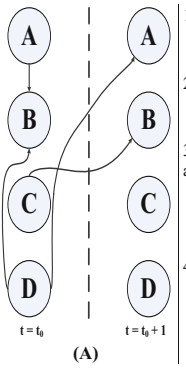
Next, we show in Theorem 1 that CCIT takes joint regulation into account while scoring and it is different than three related approaches, namely MIT [9] applied to: a Bayesian Network (which we call MIT_0); a dynamic Bayesian Network (called MIT_1); and also a naive combination of these two, where the intra and inter-slice networks are scored independently (called MIT_{0+1}). For this, we make use of the decomposition property of MI, defined next:

Property 1. (Decomposition Property of MI) In a BN, if $Pa(X_i)$ is the parent set of a node X_i ($X_{ik} \in Pa(X_i), k = 1, \ldots s_i$), and the cardinality of the set is s_i, the following identity holds [9]:

$$MI(X_i, Pa(X_i)) = MI(X_i, X_{i1}) + \sum_{j=2}^{s_i} MI\left(X_i, X_{ij} | \left\{X_{i1}, \ldots, X_{i(j-1)}\right\}\right) \tag{5}$$

Theorem 1. *CCIT scores intra and inter-slice arcs concurrently, and is different from MIT_0, MIT_1 and MIT_{0+1} since it takes into account the fact that multiple regulators may regulate a gene simultaneously, rather than in an isolated manner.*

Proof. We prove by showing a counterexample, using the network in Fig. 4(A). We apply our metric along with the three other techniques on the network,



1. Application of MIT in a BN based framework:

$$S_{MIT_0} = 2N.MI(B,\{A^0,D^0\}) - (\chi_{\alpha,4} + \chi_{\alpha,12}) \quad (6)$$

2. Application of MIT in a DBN based framework:

$$S_{MIT_1} = 2N\{MI(B,C^1) + MI(A,D^1)\} - 2\chi_{\alpha,4} \quad (7)$$

3. A naive application of MIT in a combined BN and DBN based framework:

$$S_{MIT_{0+1}} = 2N\{MI(B,\{A^0,D^0\}) + MI(B,C^1) + MI(A,D^1)\} - (3\chi_{\alpha,4} + \chi_{\alpha,12}) \quad (8)$$

4. Our proposed scoring metric:

$$S_{CCIT} = 2N\{MI(B,\{A^0,D^0\} \cup \{C^1\}) + MI(A,D^1)\} - (3\chi_{\alpha,4} + \chi_{\alpha,12}) \quad (9)$$

(B)

Fig. 4. (A) Network used for the proof (rolled representation). (B) equations depicting how each approach will score the network in 4(A).

describe the working procedure in all these cases to show that the proposed metric indeed scores them concurrently, and finally show the difference with the other three approaches. We assume the non-trivial case where the data is supposed to be periodic (the proof is trivial otherwise). Also, we assume that all the gene expressions were discretized to 3 quantization levels. The concurrent scoring behavior of CCIT is evident from the first term in RHS of (9), as shown in Fig. 4(B). Also, inclusion of C in the parent set in the first term of the RHS of the equation exhibits the way how it achieves the objective of taking into account the biological fact that multiple regulators may regulate a gene jointly.

Considering (6) to (8) in Fig. 4(B), it is also obvious that CCIT is different from both MIT_0 and MIT_1. To show that CCIT is different from MIT_{0+1}, we consider (8) and (9). It suffices to consider whether $MI(B,\{A^0,D^0\}) + MI(B,C^1)$ is different from $MI(B,\{A^0,D^0\} \cup \{C^1\})$. Using (5), this becomes equivalent to considering whether $MI(B,\{A^0,D^0\}|C^1)$ is the same as $MI(B,\{A^0,D^0\})$, which are clearly inequal. This completes the proof. □

4 The Search Strategy

A genetic algorithm (GA), applied to explore this structure space, begins with a sample population of randomly selected network structures and their fitness calculated. Iteratively, crossovers and mutations of networks within a population are performed and the best fitting individuals are kept for future generations. During crossover, random edges from different networks are chosen and swapped. Mutation is applied on a subset of edges of every network. For our study, we incorporate the following three types of mutations: *(i)* Deleting a random edge from the network, *(ii)* Creating a random edge in the network, and *(iii)* Changing direction of a randomly selected edge.

The overall algorithm that includes the modeling of the GRN and the stochastic search of the network space using GA is shown in Table 1.

Table 1. Genetic Algorithm

1. Create initial population of network structures (100 in our case). For each individual, genes and set of parent genes are selected based on a Poisson distribution and edges are created such that the resulting network complies with the conditions listed in Section 2.
2. Evaluate each network and sort the chromosomes based on the fitness score.
   (a) Generate new population by applying crossover and mutation on the previous population. Check to see if any conditions listed in Section 2 is violated.
   (b) Evaluate each individual using the fitness function and use it to sort the individual networks.
   (c) If the best individual score has not increased for consecutive 5 times, aggregate the 5 best individuals using a majority voting scheme. Check to see if any conditions listed in Section 2 is violated.
   (d) Take best individuals from the two populations and create the population of elite individuals for next generation.
3. Repeat steps a) - d) until the stopping criteria (400 generations/no improvement in fitness for 10 consecutive generations) is reached. When the GA stops, take the best chromosome and reconstruct the final genetic network.

5 Experimental Evaluation

We evaluate our method using both: synthetic network and a real-life biological network of *Saccharomyces cerevisiae* (yeast).We used the Persist Algorithm [10] to discretize continuous data into 3 levels. The value of the confidence level (α) used was 0.90. We applied four widely known performance measures, namely Sensitivity (Se), Specificity (Sp), Precision (Pr) and F-Score (F) and compared our method with other recent as well as traditional methods.

5.1 Synthetic Network

Synthetic Network having both Instantaneous and Time-Delayed Interactions. As a first step towards evaluating our approach, we employ a 9 node network shown in Fig. 5. We used N = 30, 50, 100 and 200 samples and generated 5 datasets in each case using random multinomial CPDs sampled from a Dirichlet, with hyper-parameters chosen using the method of [11]. The results are shown in Table 2. It is observed that both DBN(DP) [5] and our method outperform MIT_{0+1}, although our method is less data intensive, and performs better than DBN(DP) [5] when the number of samples is low.

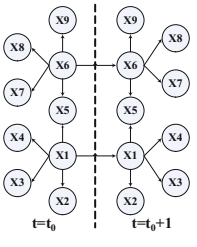


Fig. 5. 9-node synthetic network

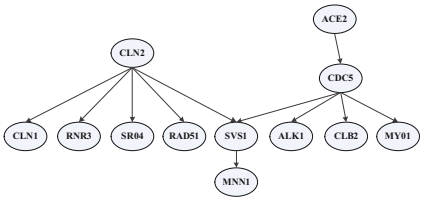


Fig. 6. Yeast cell cycle subnetwork [12]

Probabilistic Network from Yeast. We use a subnetwork from the yeast cell cycle, shown in Fig. 6, taken from Husmeier et al. [12]. The network consists of 12 genes and 11 interactions. For each interaction, we randomly assigned a

Table 2. Performance comparison of proposed method with, DBN(DP) and MIT_{0+1} on the 9-node synthetic network

	N=30			N=50			N=100			N=200		
	Se	***Sp***	***F***	***Se***	***Sp***	***F***	***Se***	***Sp***	***F***	***Se***	***Sp***	***F***
Proposed Method	0.18 ± 0.1	0.99± 0.0	0.28± 0.15	0.50± 0.14	0.91± 0.04	0.36± 0.13	0.54± 0.05	0.93± 0.02	0.42± 0.05	0.56± 0.11	0.99± 0.01	0.65± 0.14
DBN (DP)	0.16± 0.08	0.99± 0.01	0.25± 0.13	0.22± 0.2	0.99± 0.0	0.32± 0.2	0.52± 0.04	1.0± 0.0	0.67± 0.05	0.58± 0.08	1.0± 0.0	0.72± 0.06
MIT_{0+1}	0.18± 0.08	0.89± 0.07	0.17± 0.1	0.26± 0.16	0.90± 0.03	0.19± 0.1	0.36± 0.13	0.88± 0.04	0.25± 0.15	0.48± 0.04	0.95± 0.03	0.45± 0.08

regulation order of 0-3. We used two different conditional probabilities for the interactions between the genes (see [12] for details about the parameters). Eight confounder nodes were also added, making the total number of nodes 20.

We used 30, 50 and 100 samples, generated 5 datasets in each case and compared our approach with two other DBN based methods, namely BANJO [13] and BNFinder [14]. While calculating performance measures for these methods, we ignored the exact orders for the time-delayed interactions in the target network. Due to scalability issues, we did not apply DBN(DP) [5] to this network. The results are shown in Table 3, where we observe that our method outperforms the other two. This points to the strength of our method in discovering complex interaction scenarios where multiple regulators may jointly regulate target genes with varying time-delays.

Table 3. Performance comparison of proposed method with, BANJO and BNFinder on the yeast subnetwork

	N=30				N=50				N=100			
	Se	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***
Proposed Method	0.73± 0.22	0.998± 0.0007	0.82± 0.09	0.75± 0.1	0.82± 0.1	0.999± 0.0010	0.85± 0.08	0.83± 0.09	0.86± 0.08	0.999± 0.0010	0.87± 0.06	0.86± 0.06
BANJO	0.51± 0.08	0.987± 0.01	0.49± 0.2	0.46± 0.15	0.55± 0.09	0.993± 0.0049	0.57± 0.23	0.55± 0.16	0.60± 0.08	0.995± 0.0014	0.61± 0.09	0.61± 0.08
BNFinder +MDL	0.51± 0.08	0.996± 0.0006	0.63± 0.07	0.56± 0.08	0.60± 0.05	0.996± 0.0022	0.68± 0.15	0.63± 0.09	0.65± 0.0	0.996± 0.0	0.69± 0.04	0.67± 0.02
BNFinder +BDe	0.53± 0.04	0.996± 0.0006	0.68± 0.02	0.59± 0.02	0.62± 0.04	0.997± 0.0019	0.74± 0.13	0.67± 0.06	0.69± 0.08	0.997± 0.0007	0.74± 0.06	0.72± 0.07

5.2 Real-Life Biological Data

To validate our method with a real-life biological gene regulatory network, we investigate a recent network, called IRMA, of the yeast *Saccharomyces cerevisiae* [15]. The network is composed of five genes regulating each other; it is also negligibly affected by endogenous genes. There are two sets of gene profiles called Switch ON and Switch OFF for this network, each containing 16 and 21 time points, respectively. A 'simplified' network, ignoring some internal protein level interactions, is also reported in [15]. To compare our reconstruction method, we consider 4 recent methods, namely, TDARACNE [16], NIR & TSNI [17], BANJO [13] and ARACNE [18].

IRMA ON Dataset. The performance comparison amongst various method based on the ON dataset is shown in Table 4. The average and standard deviation

correspond to five different runs of the GA. We observe that our method achieves good precision value as well as very high specificity. The *Se* and F-score measures are also comparable with the other methods.

Table 4. Performance comparison based on IRMA ON dataset

	Original Network				Simplified Network			
	Se	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***
Proposed Method	0.53± 0.1	0.90± 0.05	0.73± 0.09	0.61± 0.09	0.60± 0.1	0.95± 0.03	0.71± 0.13	0.65± 0.14
TDARACNE	0.63	0.88	0.71	0.67	0.67	0.90	0.80	0.73
NIR & TSNI	0.50	0.94	0.80	0.62	0.67	1	1	0.80
BANJO	0.25	0.76	0.33	0.27	0.50	0.70	0.50	0.50
ARACNE	0.60	-	0.50	0.54	0.50	-	0.50	0.50

IRMA OFF Dataset. Due to the lack of 'stimulus', it is comparatively difficult to reconstruct the exact network from the OFF dataset [16]. As a result, the overall performances of all the algorithms suffer to some extent. The comparison is shown in Table 5. Again we observe that our method reconstructs the gene network with very high precision. Specificity is also quite high, implying that the inference of false positives is low.

Table 5. Performance comparison based on IRMA OFF dataset

	Original Network				Simplified Network			
	Se	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***
Proposed Method	0.50± 0.0	0.89± 0.03	0.70± 0.05	0.58± 0.02	0.33± 0.0	0.94± 0.03	0.64± 0.08	0.40± 0.0
TDARACNE	0.60	-	0.37	0.46	0.75	-	0.50	0.60
NIR & TSNI	0.38	0.88	0.60	0.47	0.50	0.90	0.75	0.60
BANJO	0.38	0.88	0.60	0.46	0.33	0.90	0.67	0.44
ARACNE	0.33	-	0.25	0.28	0.60	-	0.50	0.54

6 Conclusion

In this paper, we introduce a framework that can simultaneously represent instantaneous and time-delayed genetic interactions. Incorporating this framework, we implement a *score+search* based GRN reconstruction algorithm using a novel scoring metric that supports the biological truth that some genes may co-regulate other genes with different orders of regulation. Experiments have been performed on different synthetic networks of varying complexities and also on real-life biological networks. Our method shows improved performance compared to other recent methods, both in terms of reconstruction accuracy and number of false predictions, at the same time maintaining comparable or better true predictions. Currently we are focusing our research on increasing the computational efficiency of the approach and its application for inferring large gene networks.

Acknowledgments. This research is a part of the larger project on genetic network modeling supported by Monash University and Australia-India Strategic Research Fund.

References

1. Ram, R., Chetty, M., Dix, T.: Causal Modeling of Gene Regulatory Network. In: Proc. IEEE CIBCB (CIBCB 2006), pp. 1–8. IEEE (2006)
2. Zou, M., Conzen, S.: A new dynamic bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. Bioinformatics 21(1), 71 (2005)
3. de Campos, C., Ji, Q.: Efficient Structure Learning of Bayesian Networks using Constraints. Journal of Machine Learning Research 12, 663–689 (2011)
4. Friedman, N., Murphy, K., Russell, S.: Learning the structure of dynamic probabilistic networks. In: Proc. UAI (UAI 1998), pp. 139–147. Citeseer (1998)
5. Eaton, D., Murphy, K.: Bayesian structure learning using dynamic programming and MCMC. In: Proc. UAI (UAI 2007) (2007)
6. Cho, R., Campbell, M., et al.: A genome-wide transcriptional analysis of the mitotic cell cycle. Molecular cell 2(1), 65–73 (1998)
7. Xing, Z., Wu, D.: Modeling multiple time units delayed gene regulatory network using dynamic Bayesian network. In: Proc. ICDM- Workshops, pp. 190–195. IEEE (2006)
8. Kullback, S.: Information theory and statistics. Wiley (1968)
9. de Campos, L.: A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. The Journal of Machine Learning Research 7, 2149–2187 (2006)
10. Morchen, F., Ultsch, A.: Optimizing time series discretization for knowledge discovery. In: Proc. ACM SIGKDD, pp. 660–665. ACM (2005)
11. Chickering, D., Meek, C.: Finding optimal bayesian networks. In: Proc. UAI (2002)
12. Husmeier, D.: Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. Bioinformatics 19(17), 2271 (2003)
13. Yu, J., Smith, V., Wang, P., Hartemink, A., Jarvis, E.: Advances to Bayesian network inference for generating causal networks from observational biological data. Bioinformatics 20(18), 3594 (2004)
14. Wilczyński, B., Dojer, N.: BNFinder: exact and efficient method for learning Bayesian networks. Bioinformatics 25(2), 286 (2009)
15. Cantone, I., Marucci, L., et al.: A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. Cell 137(1), 172–181 (2009)
16. Zoppoli, P., Morganella, S., Ceccarelli, M.: TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach. BMC Bioinformatics 11(1), 154 (2010)
17. Della Gatta, G., Bansal, M., et al.: Direct targets of the TRP63 transcription factor revealed by a combination of gene expression profiling and reverse engineering. Genome Research 18(6), 939 (2008)
18. Margolin, A., Nemenman, I., et al.: ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 7(suppl. 1), S7 (2006)

Resource Allocation and Scheduling of Multiple Composite Web Services in Cloud Computing Using Cooperative Coevolution Genetic Algorithm

Lifeng Ai[1,2], Maolin Tang[1], and Colin Fidge[1]

[1] Queensland University of Technology, 2 George Street, Brisbane, 4001, Australia
[2] Vancl Research Laboratory, 59 Middle East 3rd Ring Road, Beijing, 100022, China
{l.ai,m.tang,c.fidge}@qut.edu.au

Abstract. In cloud computing, resource allocation and scheduling of multiple composite web services is an important and challenging problem. This is especially so in a hybrid cloud where there may be some low-cost resources available from private clouds and some high-cost resources from public clouds. Meeting this challenge involves two classical computational problems: one is assigning resources to each of the tasks in the composite web services; the other is scheduling the allocated resources when each resource may be used by multiple tasks at different points of time. In addition, Quality-of-Service (QoS) issues, such as execution time and running costs, must be considered in the resource allocation and scheduling problem. Here we present a Cooperative Coevolutionary Genetic Algorithm (CCGA) to solve the deadline-constrained resource allocation and scheduling problem for multiple composite web services. Experimental results show that our CCGA is both efficient and scalable.

Keywords: Cooperative coevolution, web service, cloud computing.

1 Introduction

Cloud computing is a new Internet-based computing paradigm whereby a pool of computational resources, deployed as web services, are provided on demand over the Internet, in the same manner as public utilities. Recently, cloud computing has become popular because it brings many cost and efficiency benefits to enterprises when they build their own web service-based applications.

When an enterprise builds a new web service-based application, it can use published web services in both *private clouds* and *public clouds*, rather than developing them from scratch. In this paper, private clouds refer to internal data centres owned by an enterprise, and public clouds refer to public data centres that are accessible to the public. A composite web service built by an enterprise is usually composed of multiple component web services, some of which may be provided by the private cloud of the enterprise itself and others which may be provided in a public cloud maintained by an external supplier. Such a computing environment is called a *hybrid cloud*.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 258–267, 2011.
© Springer-Verlag Berlin Heidelberg 2011

The component web service *allocation problem* of interest here is based on the following assumptions. Component web services provided by private and public clouds may have the same functionality, but different Quality-of-Service (QoS) values, such as response time and cost. In addition, in a private cloud a component web service may have a limited number of instances, each of which may have different QoS values. In public clouds, with greater computational resources at their disposal, a component web service may have a large number of instances, with identical QoS values. However, the QoS values of service instances in different public clouds may vary. There may be many composite web services in an enterprise. Each of the tasks comprising a composite web service needs to be allocated an instance of a component web service. A single instance of a component web service may be allocated to more than one task in a set of composite web services, as long as it is used at different points of time.

In addition, we are concerned with the component web service *scheduling problem*. In order to maximise the utilisation of available component web services in private clouds, and minimise the cost of using component web services in public clouds, allocated component web service instances should only be used for a short period of time. This requires scheduling the allocated component web service instances efficiently.

There are two typical QoS-based component web service allocation and scheduling problems in cloud computing. One is the *deadline-constrained* resource allocation and scheduling problem, which involves finding a cloud service allocation and scheduling plan that minimises the total cost of the composite web service, while satisfying given response time constraints for each of the composite web services. The other is the *cost-constrained* resource allocation and scheduling problem, which requires finding a cloud service allocation and scheduling plan which minimises the total response times of all the composite web services, while satisfying a total cost constraint.

In previous work [1], we presented a random-key genetic algorithm (RGA) [2] for the constrained resource allocation and scheduling problems and used experimental results to show that our RGA was scalable and could find an acceptable, but not necessarily optimal, solution for all the problems tested. In this paper we aim to improve the quality of the solutions found by applying a cooperative coevolutionary genetic algorithm (CCGA) [3,4,5] to the deadline-constrained resource allocation and scheduling problem.

2 Problem Definition

Based on the requirements introduced in the previous section, the *deadline-constrained resource allocation and scheduling problem* can be formulated as follows.

Inputs

1. A set of composite web services $W = \{W_1, W_2, \ldots, W_n\}$, where n is the number of composite web services. Each composite web service consists of

several abstract web services. We define $O_i = \{o_{i,1}, o_{i,2}, \ldots, o_{i,n_i}\}$ as the abstract web services set for composite web service W_i, where n_i is the number of abstract web services contained in composite web service W_i.
2. A set of candidate cloud services $S_{i,j}$ for each abstract web service $o_{i,j}$, where $S_{i,j} = S^u_{i,j} \cup S^v_{i,j}$, and $S^v_{i,j} = \{S^v_{i,j,1}, S^v_{i,j,2}, \ldots, S^v_{i,j,\ell}\}$ denotes an entire set of ℓ private cloud service candidates for abstract web service $o_{i,j}$, and $S^u_{i,j} = \{S^u_{i,j,1}, S^u_{i,j,2}, \ldots, S^u_{i,j,m}\}$ denotes an entire set of m public cloud service candidates for abstract web service $o_{i,j}$.
3. A response time and price for each public cloud service $S^u_{i,j,k}$, denoted by $t^u_{i,j,k}$ and $c^u_{i,j,k}$ respectively, and a response time and price for each private cloud service $S^v_{i,j,k}$, denoted by $t^v_{i,j,k}$ and $c^v_{i,j,k}$ respectively.

Output

1. An allocation and scheduling plan $X = \{X_i \mid i = 1, 2, \ldots, n\}$, such that the total cost of X, i.e., $Cost(X) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} Cost(M_{i,j})$, is minimal, where $X_i = \{(M_{i,1}, F_{i,1}), (M_{i,2}, F_{i,2}), \ldots, (M_{i,n_i}, F_{i,n_i})\}$ denotes an allocation and scheduling plan for composite web service W_i, $M_{i,j}$ represents the selected cloud service for abstract web service $o_{i,j}$, and $F_{i,j}$ stands for the finishing time of $M_{i,j}$.

Constraints

1. All the finishing-time precedence requirements between the abstract web services are satisfied, that is, $F_{i,k} \leq F_{i,j} - d_{i,j}$, where $j = 1, \ldots, n_i$, and $k \in Pre_{i,j}$, where $Pre_{i,j}$ denotes the set of all abstract web services that must execute before the abstract web service $o_{i,j}$.
2. All the resource limitations are respected, that is, $\sum_{j \in A(t)} r_{j,m} \leq 1$, where $m \in S^v_{i,j}$ and $A(t)$ denotes the entire set of abstract web services being used at time t. Let $r_{j,m} = 1$ if abstract web service j requires private cloud service m in order to execute and $r_{j,m} = 0$ otherwise. This constraint guarantees that each private cloud service can only serve at most one abstract web service at a time.
3. The deadline constraint for each composite web service is satisfied, that is, $F_{i,n_i} \leq d_i$, such that $i = 1, \ldots, n$, where d_i denotes the deadline promised to the customer for composite web service W_i, and F_{i,n_i} is the finishing time of the last abstract service of composite web service W_i, that is, the overall execution time of the composite web service W_i.

3 A Cooperative Coevolutionary Genetic Algorithm

Our Cooperative Coevolutionary Genetic Algorithm is based on Potter and De Jong's model [3]. In their approach several *species*, or *subpopulations*, coevolve together. Each individual in a subpopulation constitutes a partial solution to the problem, and the combination of an individual from all the subpopulations forms a complete solution to the problem. The subpopulations of the CCGA

evolve independently in order to improve the individuals. Periodically, they interact with each other to acquire feedback on how well they are cooperatively solving the problem. In order to use the cooperative coevolutionary model, two major issues must be addressed, *problem decomposition* and *interaction between subpopulations*, which are discussed in detail below.

3.1 Problem Decomposition

Problem composition can be either static, where the entire problem is partitioned in advance and the number of subpopulations is fixed, or dynamic, where the number of subpopulations is adjusted during the calculation time. Since the problem studied here can be naturally decomposed into a fixed number of subproblems beforehand, the problem decomposition adopted by our CCGA is static.

Essentially our problem is to find a resource allocation scheduling solution for multiple composite web services. Thus, we define the problem of finding a resource allocation and scheduling solution for each of the composite web services as a subproblem. Therefore, the CCGA has n subpopulations, where n is the total number of composite web services involved. Each subpopulation is responsible for solving one subproblem and the n subpopulations interact with each other as the n composite web services compete for resources.

3.2 Interaction between Subpopulations

In our Cooperative Coevolutionary Genetic Algorithm, interactions between subpopulations occur when evaluating the fitness of an individual in a subpopulation. The fitness value of a particular individual in a population is an estimate of how well it cooperates with other species to produce good solutions. Guided by the fitness value, subpopulations work cooperatively to solve the problem. This interaction between the sub-populations involves the following two issues.

1. *Collaborator selection*, i.e., selecting collaborator subcomponents from each of the other subpopulations, and assembling the subcomponents with the current individual being evaluated to form a complete solution. There are many ways of selecting collaborators [6]. In our CCGA, we use the most popular one, choosing the best individuals from the other subpopulations, and combine them with the current individual to form a complete solution. This is the so-called *greedy collaborator selection method* [6].
2. *Credit assignment*, i.e., assigning credit to the individual. This is based on the principle that the higher the fitness value the complete solution has—constructed by the above collaborator selection method—the more credit the individual will obtain. The fitness function is defined by Equations 1 to 3 below. By doing so, in the following evolving rounds, an individual resulting in better cooperation with its collaborators will be more likely to survive. In other words, this credit assignment method can enforce the evolution of each population towards a better direction for solving the problem.

$$Fitness(X) = \begin{cases} F_{Max}^{Cost}/F_{obj}(X), & \text{if } V(X) \leq 1; \\ 1/V(X), & \text{otherwise.} \end{cases} \quad (1)$$

$$V(X) = \prod_{i=1}^{n} (V_i(X)) \quad (2)$$

$$V_i(X) = \begin{cases} F_{i,n_i}/d_i, & \text{if } F_{i,n_i} > d_i; \\ 1, & \text{otherwise.} \end{cases} \quad (3)$$

In Equation 1, condition $V(X) \leq 1$ means there is no constraint violation. Conversely, $V(X) > 1$ means some constraints are violated, and the larger the value of $V(X)$, the higher the degree of constraint violation. F_{Max}^{Cost} is the worst $F_{obj}(X)$, namely the maximal total cost, among all feasible individuals in a current generation. Ratio $F_{Max}^{Cost}/F_{obj}(X)$ is used to scale the fitness value of all feasible solutions into range $[1, \infty)$. Using Equations 1 to 3, we can guarantee that the fitness of all feasible solutions in a generation are better than the fitness of all infeasible solutions. In addition, the lower the total cost for a feasible solution, the better fitness the solution will have. The higher number of constraints that are violated by an infeasible solution, the worse fitness the solution will have.

3.3 Algorithm Description

Algorithm 1 summarises our Cooperative Coevolutionary Genetic Algorithm. Step 1 initialises all the subpopulations. Steps 2 to 7 evaluate the fitness of each individual in the initial subpopulations. This is done in two steps. The first step combines the individual $indiv[i][j]$ ($indiv[i][j]$ denotes the j th individual in the ith subpopulation in the CCGA) with the jth individual from each of the other subpopulations to form a complete solution c to the problem, and the second step calculates the fitness value of the solution c using the fitness function defined by Equation 1.

Steps 8 to 18 are the co-evolution rounds for the N subpopulations. In each round, the N subpopulations evolve one by one from the 1st to the Nth. When evolving a subpopulation $SubPop[i]$, where $1 \leq i \leq N$, we use the same selection, crossover and mutation operators as used in our previously-described random-key genetic algorithm (RGA) [1]. However, the fitness evaluation used in the CCGA is different from that used in the RGA. In the CCGA, we use the aforementioned collaborator selection strategy and the credit assignment method to evaluate the fitness of an individual. The cooperative co-evolution process is repeated until certain termination criteria are satisfied, specific to the application (e.g., a certain number of rounds or a fixed time limit).

4 Experimental Results

Experiments were conducted to evaluate the scalability and effectiveness of our CCGA for the resource allocation and scheduling problem by comparing it with

Algorithm 1. Our cooperative coevolutionary genetic algorithm

```
1  Construct N sets of initial populations, SubPop[i], i = 1, 2, ..., N
2  for i ← 1 to N do
3  |  foreach individual indiv[i][j] of the subpopulation SubPop[i] do
4  |  |  c ← SelectPartnersBySamePosition(j)
5  |  |  indiv[i][j].Fitness ← FitnessFunc (c)
6  |  end
7  end
8  while termination condition is not true do
9  |  for i ← 1 to N do
10 |  |  Select fit individuals in SubPop[i] for reproduction
11 |  |  Apply the crossover operator to generate new offspring for SubPop[i]
12 |  |  apply the mutation operator to offspring
13 |  |  foreach individual indiv[i][j] of the subpopulation SubPop[i] do
14 |  |  |  c ← SelectPartnersByBestFitness
15 |  |  |  indiv[i][j].Fitness ← FitnessFunc (c)
16 |  |  end
17 |  end
18 end
```

our previous RGA [1]. Both algorithms were implemented in Microsoft Visual C♯, and the experiments were conducted on a desktop computer with a 2.33 GHz Intel Core 2 Duo CPU and a 1.95 GB RAM. The population sizes of the RGA and the CCGA were 200 and 100, respectively. The probabilities for crossover and mutation in both the RGA and the CCGA were 0.85 and 0.15, respectively. The termination condition used in the RGA was "no improvement in 40 consecutive generations", while the termination condition used in the CCGA was "no improvement in 20 consecutive generations". These parameters were obtained through trials on randomly generated test problems. The parameters that led to the best performance in the trials were selected.

The scalability and effectiveness of the CCGA and RGA were tested on a number of problem instances with different sizes. Problem size is determined by three factors: the number of composite web services involved in the problem, the number of abstract web services in each composite web service, and the number of candidate cloud services for each abstract service. We constructed three types of problems, each designed to evaluate how one of the three factors affects the computation time and solution quality of the algorithms.

4.1 Experiments on the Number of Composite Web Services

This experiment evaluated how the number of composite web services affects the computation time and solution quality of the algorithms. In this experiment, we also compared the algorithms' convergence speeds. Considering the stochastic nature of the two algorithms, we ran both ten times on each of the randomly generated test problems with a different number of composite web services. In

this experiment, the number of composite web services in the test problems ranged from 5 to 25 with an increment of 5. The deadline constraints for the five test problems were 59.4, 58.5, 58.8, 59.2 and 59.8 minutes, respectively. Because of space limitations, the five test problems are not given in this paper, but they can be found elsewhere [1].

The experimental results are presented in Table 1. It can be seen that both algorithms always found a feasible solution to each of the test problems, but that the solutions found by the CCGA are consistently better than those found by the RGA. For example, for the test problem with five composite web services, the average cost of the solutions found by the RGA of ten times run was $103, while the average cost of the solutions found by the CCGA was only $79. Thus, $24 can be saved by using the CCGA on average.

Table 1. Comparison of the algorithms with different numbers of composite web services

No. of Composite Web Services	RGA		CCGA	
	Feasible Solution	Aver. Cost ($)	Feasible Solution	Ave. Cost ($)
5	Yes	103	Yes	79
10	Yes	171	Yes	129
15	Yes	326	Yes	251
20	Yes	486	Yes	311
25	Yes	557	Yes	400

The computation time of the two algorithms as the number of composite web services increases is shown in Figure 1. The computation time of the RGA increased close to linearly from 25.4 to 226.9 seconds, while the computation time of the CCGA increased super-linearly from 6.8 to 261.5 seconds as the number of composite web services increased from 5 to 25. Although the CCGA is not as scalable as the RGA there is little overall difference between the two algorithms for problems of this size, and a single web service would not normally comprise very large numbers of components.

4.2 Experiments on the Number of Abstract Web Services

This experiment evaluated how the number of abstract web services in each composite web service affects the computation time and solution quality of the algorithms. In this experiment, we randomly generated five test problems. The number of abstract web services in the five test problems ranged from 5 to 25 with an increment of 5. The deadline constraints for the test problems were 26.8, 59.1, 89.8, 117.6 and 153.1 minutes, respectively. The quality of the solutions found by the two algorithms for each of the test problems is shown in Table 2. Once again both algorithms always found feasible solutions, and the CCGA always found better solutions than the RGA.

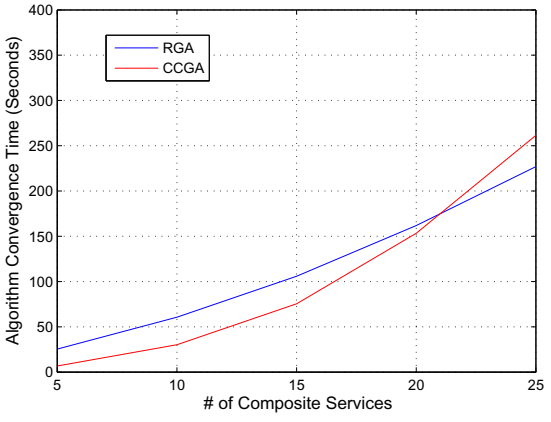


Fig. 1. Number of composite web services versus computation time for both algorithms

Table 2. Comparison of the algorithms with different numbers of abstract web services

No. of	RGA		CCGA	
Abstract Services	Feasible Solution	Ave. Cost ($)	Feasible Solution	Ave. Cost ($)
5	Yes	105	Yes	81
10	Yes	220	Yes	145
15	Yes	336	Yes	259
20	Yes	458	Yes	322
25	Yes	604	Yes	463

The computation times of the two algorithms as the number of abstract web services involved in each composite web service increases are displayed in Figure 2. The Random-key GA's computation time increased linearly from 29.8 to 152.3 seconds and the Cooperative Coevolutionary GA's computation time increased linearly from 14.8 to 72.1 seconds as the number of abstract web services involved in the each composite web service grew from 5 to 25. On this occasion the CCGA clearly outperformed the RGA.

4.3 Experiments on the Number of Candidate Cloud Services

This experiment examined how the number of candidate cloud services for each of the abstract web services affects the computation time and solution quality of the algorithms. In this experiment, we randomly generated five test problems. The number of candidate cloud services in the five test problems ranged from 5 to 25 with an increment of 5, and the deadline constraints for the test problems were 26.8, 26.8, 26.8, 26.8 and 26.8 minutes, respectively. Table 3 shows that yet again both algorithms always found feasible solutions, with those produced by the CCGA being better than those produced by the RGA.

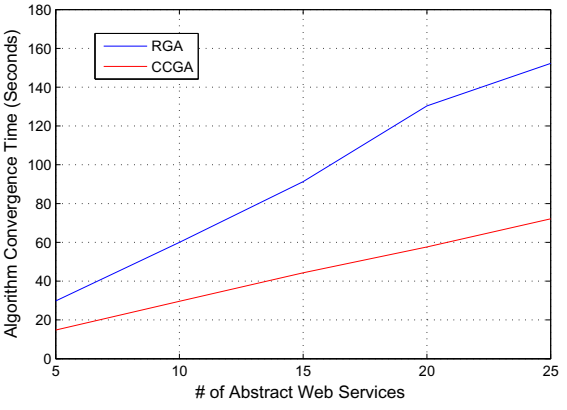


Fig. 2. Number of abstract web services versus computation time for both algorithms

Table 3. Comparison of the algorithms with different numbers of candidate cloud services for each abstract service

No. of Candidate	RGA		CCGA	
Web Services	Feasible Solution	Ave. Cost ($)	Feasible Solution	Ave. Cost ($)
5	Yes	144	Yes	130
10	Yes	142	Yes	131
15	Yes	140	Yes	130
20	Yes	141	Yes	130
25	Yes	142	Yes	130

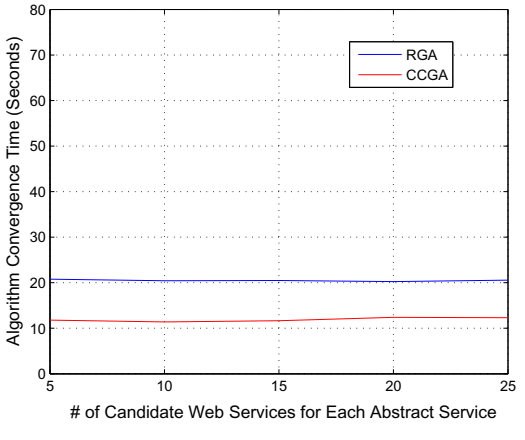


Fig. 3. Number of candidate cloud services versus computation time for both algorithms

Figure 3 shows the relationship between the number of candidate cloud services for each abstract web service and the algorithms' computation times.

Increasing the number of candidate cloud services had no significant effect on either algorithm, and the computation time of the CCGA was again much better than that of the RGA.

5 Conclusion and Future Work

We have presented a Cooperative Coevolutionary Genetic Algorithm which solves the deadline-constrained cloud service allocation and scheduling problem for multiple composite web services on hybrid clouds. To evaluate the efficiency and scalability of the algorithm, we implemented it and compared it with our previously-published Random-key Genetic Algorithm for the same problem. Experimental results showed that the CCGA always found better solutions than the RGA, and that the CCGA scaled up well when the problem size increased.

The performance of the new algorithm depends on the collaborator selection strategy and the credit assignment method used. Therefore, in future work we will look at alternative collaborator selection and credit assignment methods to further improve the performance of the algorithm.

Acknowledgement. This research was carried out as part of the activities of, and funded by, the Cooperative Research Centre for Spatial Information (CRC-SI) through the Australian Government's CRC Programme (Department of Innovation, Industry, Science and Research).

References

1. Ai, L., Tang, M., Fidge, C.: QoS-oriented resource allocation and scheduling of multiple composite web services in a hybrid cloud using a random-key genetic algorithm. Australian Journal of Intelligent Information Processing Systems 12(1), 29–34 (2010)
2. Bean, J.C.: Genetic algorithms and random keys for sequencing and optimization. ORSA Journal on Computing 6(2), 154–160 (1994)
3. Potter, M.A., De Jong, K.A.: Cooperative coevolution: An architecture for evolving coadapted subcomponents. Evolutionary Computation 8(1), 1–29 (2000)
4. Ray, T., Yao, X.: A cooperative coevolutionary algorithm with correlation based adaptive variable partitioning. In: Proceeding of IEEE Congress on Evolutionary Computation, pp. 983–989 (2009)
5. Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. Information Sciences 178(15), 2985–2999 (2008)
6. Wiegand, R.P., Liles, W.C., De Jong, K.A.: An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1235–1242 (2001)

Image Classification Based on Weighted Topics

Yunqiang Liu[1] and Vicent Caselles[2]

[1] Barcelona Media - Innovation Center, Barcelona, Spain
liuyunq@yahoo.com
[2] Universitat Pompeu Fabra, Barcelona, Spain
vicent.caselles@upf.edu

Abstract. Probabilistic topic models have been applied to image classification and permit to obtain good results. However, these methods assumed that all topics have an equal contribution to classification. We propose a weight learning approach for identifying the discriminative power of each topic. The weights are employed to define the similarity distance for the subsequent classifier, e.g. KNN or SVM. Experiments show that the proposed method performs effectively for image classification.

Keywords: Image classification, pLSA, topics, learning weights.

1 Introduction

Image classification, i.e. analyzing and classifying the images into semantically meaningful categories, is a challenging and interesting research topic. The bag of words (BoW) technique [1], has demonstrated remarkable performance for image classification. Under the BoW model, the image is represented as a histogram of visual words, which are often derived by vector quantizing automatically extracted local region descriptors.

The BoW approach is further improved by a probabilistic semantic topic model, e.g. probabilistic latent semantic analysis (pLSA) [2], which introduces intermediate latent topics over visual words [2,3,4]. The topic model was originally developed for topic discovery in text document analysis. When the topic model is applied to images, it is able to discover latent semantic topics in the images based on the co-occurrence distribution of visual words. Usually, the topics, which are used to represent the content of an image, are detected based on the underlying probabilistic model, and image categorization is carried out by taking the topic distribution as the input feature. Typically, the k-nearest neighbor classifier (KNN) [5] or the support vector machine (SVM) [6] based on the Euclidean distance are adopted for classification after topic discovery.

In [7], continuous vocabulary models are proposed to extend the pLSA model, so that visual words are modeled as continuous feature vector distributions rather than crudely quantized high-dimensional descriptors. Considering that the Expectation Maximization algorithm in pLSA model is sensitive to the initialization, Lu et al. [8] provided a good initial estimation using rival penalized competitive learning.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 268–275, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Most of these methods assume that all semantic topics have equal importance in the task of image classification. However, some topics can be more discriminative than others because they are more informative for classification. The discriminative power of each topic can be estimated from a training set with labeled images. This paper tries to exploit discriminatory information of topics based on the intuition that the weighted topics representation of images in the same category should more similar than that of images from different categories. This idea is closely related to the distance metric learning approaches which are mainly designed for clustering and KNN classification [5]. Xing et al. [9] learn a distance metric for clustering by minimizing the distances between similarly labeled data while maximizing the distances between differently labeled data. Domeniconi et al. [10] use the decision boundaries of SVMs to induce a locally adaptive distance metric for KNN classification. Weinberger et al. [11] propose a large margin nearest neighbor (LMNN) classification approach by formulating the metric learning problem in a large margin setting for KNN classification.

In this paper, we introduce a weight learning approach for identifying the discriminative power of each topic. The weights are trained so that the weighted topics representations of images from different categories are separated with a large margin. The weights are employed to define the weighted Euclidean distance for the subsequent classifier, e.g. KNN or SVM. The use of a weighted Euclidean distance can equivalently be interpreted as taking a linear transformation of the input space before applying the classifier using Euclidean distances. The proposed weighted topics representation of images has a higher discriminative power in classification tasks. Experiments show that the proposed method can perform quite effectively for image classification.

2 Classification Based on Weighted Topics

We describe in this section the weighted topics method for image classification. First, the image is represented using the bag of words model. Then we briefly review the pLSA method. And finally, we introduce the method to learn the weights for the classifier.

2.1 Image Representation

Dense image feature sampling is employed since comparative results have shown that using a dense set of keypoints works better than sparsely detected keypoints in many computer vision applications [2]. In this work, each image is divided into equivalent blocks on a regular grid with spacing d. The set of grid points are taken as keypoints, each with a circular support area of radius r. Each support area can be taken as a local patch. The patches are overlapped when $d < 2r$. Each patch is described by a descriptor like SIFT (Scale-Invariant Feature Transform) [12]. Then a visual vocabulary is built-up by vector quantizing the descriptors using a clustering algorithm such as K-means. Each resulting cluster corresponds to a visual word. With the vocabulary, each descriptor is assigned to its nearest visual word in the visual vocabulary. After mapping keypoints into visual

words, the word occurrences are counted, and each image is then represented as a term-frequency vector whose coordinates are the counts of each visual word in the image, i.e. as a histogram of visual words. These term-frequency vectors associated to images constitute the co-occurrence matrix.

2.2 pLSA Model for Image Analysis

The pLSA model is used to discover topics in an image based on the bag of words image representation.

Assume that we are given a collection of images $D = \{d_1, d_2, ..., d_N\}$, with words from a visual vocabulary $W = \{w_1, w_2, ..., w_V\}$. Given $n(w_i, d_j)$, the number of occurrences of word i in image d_j for all the images in the training database, pLSA uses a finite number of hidden topics $Z = \{z_1, z_2, ..., z_K\}$ to model the co-occurrence of visual words inside and across images. Each image is characterized as a mixture of hidden topics. The probability of word w_i in image d_j is defined by the following model:

$$P(w_i, d_j) = P(d_j) \sum_k P(z_k|d_j) P(w_i|z_k), \tag{1}$$

where $P(d_j)$ is the prior probability of picking image d_j, which is usually set as a uniform distribution, $P(z_k|d_j)$ is the probability of selecting a hidden topic depending on the current image and $P(w_i|z_k)$ is the conditional probability of a specific word w_i conditioned by the unobserved topic variable z_k. The model parameters $P(z_k|d_j)$ and $P(w_i|z_k)$ are estimated by maximizing the following log-likelihood objective function using the Expectation Maximization (EM) algorithm:

$$\ell(P) = \sum_i \sum_j n(w_i, d_j) \log P(w_i, d_j), \tag{2}$$

where P denotes the family of probabilities $P(w_i|z_k)$, $i = 1, \ldots, V$, $k = 1, \ldots, K$.

The EM algorithm estimates the parameters of pLSA model as follows:

E_step

$$P(z_k|w_i, d_j) = \frac{P(z_k|d_j) P(w_i|z_k)}{\sum_m P(z_m|d_j) P(w_i|z_m)} \tag{3}$$

M_step

$$P(w_i|z_k) = \frac{\sum_j n(w_i, d_j) P(z_k|w_i, d_j)}{\sum_m \sum_j n(w_m, d_j) P(z_k|w_m, d_j)} \tag{4}$$

$$P(z_k|d_j) = \frac{\sum_i n(w_i, d_j) P(z_k|w_i, d_j)}{\sum_m \sum_i n(w_i, d_j) P(z_m|w_i, d_j)}. \tag{5}$$

Once the model parameters are learned, we can obtain the topic distribution of each image in the training dataset. The topic distributions of test images are estimated by a fold-in technique by keeping $P(w_i|z_k)$ fixed [3].

2.3 Learning Weights for Topics

Most of pLSA based image classification methods assume that all semantic topics have equally importance for the classification task and should be equally weighted. This is implicit in the use of Euclidean distances between topics. In concrete situations, some topic may be more relevant than others and turn out to have more discriminative power for classification. The discriminative power of each topic can be estimated from a training set with labeled images. This paper tries to exploit the discriminative information of different topics based on the intuition that images in the same category should have a more similar weighted topics representation when compared to images in other categories. This behavior should be captured by using a weighted Euclidean distance between images x_i and x_j given by:

$$d_\omega(x_i, x_j) = \left\{ \sum_{m=1}^{K} \omega_m ||z_{m,i} - z_{m,j}||^2 \right\}^{\frac{1}{2}}, \tag{6}$$

where $\omega_m \geq 0$ are the weights to be learned, and $\{z_{m,i}\}_{m=1}^K$, $\{z_{m,j}\}_{m=1}^K$ are the topic representation using the pLSA model of images x_i and x_j. Each topic is described by a vector in R^q for some $q \geq 1$ and $||z||$ denotes the Euclidean norm of the vector $z \in R^q$. Thus, the complete topic space is $R^{q \times K}$.

The desired weights ω_m are trained so that images from different categories are separated with a large margin, while the distance between examples in the same category should be small. In this way, images from the same category move closer and those from different categories move away in the weighted topics image representation. Thus the weights should help to increase the separability of categories. For that the learned weights should satisfy the constraints

$$d_\omega(x_i, x_k) > d_\omega(x_i, x_j), \qquad \forall (i, j, k) \in T, \tag{7}$$

where T is the index set of triples of training examples

$$T = \{(i, j, k) : y_i = y_j, y_i \neq y_k\}, \tag{8}$$

and y_i and y_j denote the class labels of images x_i and x_j.

It is not easy to satisfy all these constraints simultaneously. For that reason one introduces slack variables ξ_{ijk} and relax the constraints (7) by

$$d_\omega(x_i, x_k)^2 - d_\omega(x_i, x_j)^2 \geq 1 - \xi_{ijk}, \qquad \forall (i, j, k) \in T. \tag{9}$$

Finally, one expects that the distance between images of the same category is small. Based on all these observations, we formulate the following constrained optimization problem:

$$\begin{aligned} &\min_{\omega, \xi_{ijk}} \sum_{(i,j) \in S} d_\omega(x_i, x_j)^2 + C \sum_{i=1}^{n} \xi_{ijk}, \\ &\text{subject to} \quad d_\omega(x_i, x_k)^2 - d_\omega(x_i, x_j)^2 \geq 1 - \xi_{ijk}, \\ &\xi_{ijk} \geq 0, \quad \forall (i, j, k) \in D, \\ &\omega_m \geq 0, \quad m = 1, ..., K, \end{aligned} \tag{10}$$

where S is the set of example pairs which belong to the same class, and C is a positive constant. As usual, the slack variables ξ_{ijk} allow a controlled violation of the constraints. A non-zero value of ξ_{ijk} allows a triple $(i, j, k) \in D$ not to meet the margin requirement at a cost proportional to ξ_{ijk}. The optimization problem (10) can be solved using standard optimization software [13]. It can be noticed that the optimization can be computationally infeasible due to the eventually very large amount of constraints (9). Notice that the unknowns enter linearly in the cost functional and in the constraints and the problem is a standard linear programming problem.

In order to reduce the memory and computational requirements, a subset of sample examples and constraints is selected. Thus, we define

$$\begin{aligned} S &= \{(i, j) : y_i = y_j, \eta_{ij} = 1\}, \\ T &= \{(i, j, k) : y_i = y_j, y_i \neq y_k, \eta_{ij} = 1, \eta_{ik} = 1\}, \end{aligned} \tag{11}$$

where η_{ij} indicates whether example j is a neighbor of image i and, at this point, neighbors are defined by a distance with equal weights such as the Euclidean distance. The constraints in (11) restrict the domain of neighboring pairs. That is, only images which are neighbor and do not share the same category label will be separated using the learned weights. On the other hand, we do not pay attention to pairs which belong to different categories and are originally separated by a large distance. This is reasonable and provides, in practice, good results for image classification. Once the weights are learned, the new weighted distance is applied in the classification step.

2.4 Classifiers with Weights

The k-nearest neighbor (KNN) is a simple yet appealing method for classification. The performance of KNN classification depends crucially on the way distances between different images are computed. Usually, the distance used is the Euclidean distance. We try to apply the learned weights into KNN classification in order to improve its performance. More specifically, the distance between two different images is measured using formula (6), instead of the standard Euclidean distance.

In SVM classification, a proper choice of the kernel function is necessary to obtain good results. In general, the kernel function determines the degree of similarity between two data vectors. Many kernel functions have been proposed. A common kernel function is the radial basis function (RBF), which measures the similarity between two vectors x_i and x_j by:

$$k_{rbf}(x_i, x_j) = \exp(-\frac{d(x_i, x_j)^2}{\gamma}), \qquad \gamma > 0, \tag{12}$$

where γ is the width of the Gaussian, and $d(x_i, x_j)$ is the distance between x_i and x_j, often defined as the Euclidean distance. With the learned weights, the distance is substituted by $d_\omega(x_i, x_j)$ given in (6). Notice in passing that we may assume that $\omega_m > 0$, otherwise we discard the corresponding topic. Then k_{rbf} is a Mercer kernel [14] (even in the topic space describing the images is taken as $R^{q \times K}$).

3 Experiments

We evaluated the weighted topics method, named as pLSA-W, for an image classification task on two public datasets: OT [15] and MSRC-2 [16]. We first describe the implementation setup. Then we compare our method with the standard pLSA-based image classification method using KNN and SVM classifiers on both datasets. For the SVM classifier, the RBF kernel is applied. The parameters such as number of neighbors in KNN and the regularization parameter c in SVM are determined using k-fold ($k = 5$) cross validation.

3.1 Experimental Setup

For the two datasets, we use only the grey level information in all the experiments, although there may be room for further improvement by including color information. First, the keypoints of each image are obtained using dense sampling, specifically, we compute keypoints on a dense grid with spacing $d = 7$ both in horizontal and vertical directions. SIFT descriptors are computed at each patch over a circular support area of radius $r = 5$.

3.2 Experimental Results

OT Dataset

OT dataset consists of a total of 2688 images from 8 different scene categories: *coast, forest, highway, insidecity, mountain, opencountry, street, tallbuilding*. We divided the images randomly into two subsets of the same size to form a training set and a test set. In this experiment, we fixed the number of topics to 25 and the visual vocabulary size to 1500. These parameters have been shown to give a good performance for this dataset [2,4]. Figure 1 shows the classification accuracy when varying the parameter k using a KNN classifier. We observe that the pLSA-W method gives better performance than the pLSA constantly, and it achieves the best classification result at $k = 11$. Table 1 shows the averaged classification results over five experiments with different random splits of the dataset.

MSRC-2 Dataset

In the experiments with MSRC-2, there are 20 classes, and 30 images per class in this dataset. We choose six classes out of them: *airplane, cow, face, car, bike, sheep*. Moreover, we divided randomly the images within each class into two groups of the same size to form a training set and a test set. We used k-fold ($k = 5$) cross validation to find the best configuration parameter for the pLSA model. In the experiment, we fix the number of visual words to 100 and optimize the number of topics. We repeat each experiment five times over different splits. Table 1 shows the averaged classification results obtained using pLSA and pLSA-W with KNN and SVM classifiers on the MSRC-2 dataset.

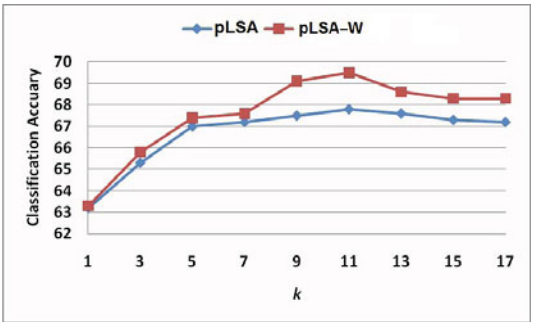


Fig. 1. Classification accuracy (%) varying the parameter k of KNN

Table 1. Classification accuracy (%)

DataSet	OT		MSRC-2	
Method	pLSA	pLSA-W	pLSA	pLSA-W
KNN	67.8	69.5	80.7	83.2
SVM	72.4	73.6	86.1	87.9

4 Conclusions

This paper proposed an image classification approach based on weighted latent semantic topics. The weights are used to identify the discriminative power of each topic. We learned the weights so that the weighted topics representation of images from different categories are separated with a large margin. The weights are then employed to define the similarity distance for the subsequent classifier, such as KNN or SVM. The use of a weighted distance makes the topic representation of images have a higher discriminative power in classification tasks than using the Euclidean distance. Experimental results demonstrated the effectiveness of the proposed method for image classification.

Acknowledgements. This work was partially funded by Mediapro through the Spanish project CENIT-2007-1012 i3media and by the Centro para el Desarrollo Tecnológico Industrial (CDTI). The authors acknowlege partial support by the EU project "2020 3D Media: Spatial Sound and Vision" under FP7-ICT. Y. Liu also acknowledges partial support from the Torres Quevedo Program from the Ministry of Science and Innovation in Spain (MICINN), co-funded by the European Social Fund (ESF). V. Caselles also acknowledges partial support by MICINN project, reference MTM2009-08171, by GRC reference 2009 SGR 773 and by "ICREA Acadèmia" prize for excellence in research funded both by the Generalitat de Catalunya.

References

1. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: Proc. ICCV, vol. 2, pp. 1470–1147 (2003)
2. Bosch, A., Zisserman, A., Muñoz, X.: Unsupervised Learning by Probabilistic Latent Semantic Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(4), 712–727 (2008)
3. Schölkopf, B., Smola, A.J.: Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning 47, 177–196 (2001)
4. Horster, E., Lienhart, R., Slaney, M.: Comparing local feature descriptors in plsa-based image models. Pattern Recognition 42, 446–455 (2008)
5. Ramanan, D., Baker, S.: Local distance functions: A taxonomy, new algorithms, and an evaluation. In: Proc. ICCV, pp. 301–308 (2009)
6. Vapnik, V.N.: Statistical learning theory. Wiley Interscience (1998)
7. Horster, E., Lienhart, R., Slaney, M.: Continuous visual vocabulary models for pLSA-based scene recognition. In: Proc. CVIR 2008, New York, pp. 319–328 (2008)
8. Lu, Z., Peng, Y., Ip, H.: Image categorization via robust pLSA. Pattern Recognition Letters 31(4), 36–43 (2010)
9. Ramanan, X.E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning with application to clustering with side-information. In: Proc. Advances in Neural Information Processing Systems, pp. 521–528 (2003)
10. Domeniconi, C., Gunopulos, D., Peng, J.: Large margin nearest neighbor classifiers. IEEE Transactions on Neural Networks 16(4), 899–909 (2005)
11. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. The Journal of Machine Learning Research 10, 207–244 (2009)
12. Lowe, G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
13. Grant, M., Boyd, S.: CVX: Matlab Software for Disciplined Convex Programming, version 1.21 (2011), http://cvxr.com/cvx
14. Schölkopf, B., Smola, A.J.: Learning with kernels. The MIT Press (2002)
15. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. International Journal of Computer Vision 42(3), 145–175 (2004)
16. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: IEEE Proc. ICCV, vol. 2, pp. 800–1807 (2005)

A Variational Statistical Framework for Object Detection

Wentao Fan[1], Nizar Bouguila[1], and Djemel Ziou[2]

[1] Concordia University, QC, Cannada
wenta_fa@encs.concordia.ca, bouguila@ciise.concordia.ca
[2] Sherbrooke University, QC, Cannada
djemel.ziou@usherbrooke.ca

Abstract. In this paper, we propose a variational framework of finite Dirichlet mixture models and apply it to the challenging problem of object detection in static images. In our approach, the detection technique is based on the notion of visual keywords by learning models for object classes. Under the proposed variational framework, the parameters and the complexity of the Dirichlet mixture model can be estimated simultaneously, in a closed-form. The performance of the proposed method is tested on challenging real-world data sets.

Keywords: Dirichlet mixture, variational learning, object detection.

1 Introduction

The detection of real-world objects poses challenging problems [1,2]. The main goal is to distinguish a given object class (e.g. car, face) from the rest of the world objects. It is very challenging because of changes in viewpoint and illumination conditions which can dramatically alter the appearance of a given object [3,4,5]. Since object detection is often the first task in many computer vision applications, many research works have been done [6,7,8,9,10,11]. Recently, several researches have adopted the bag of visual words model (see, for instance, [12,13,14]). The main idea is to represent a given object by a set of local descriptors (e.g. SIFT [15]) representing local interest points or patches. These local descriptors are then quantized into a visual vocabulary which allows the representation of a given object as a histogram of visual words.

The introduction of the notion of visual words has allowed significant progress in several computer vision applications and possibility to develop models inspired by text analysis such as pLSA [16]. The goal of this paper is to propose an object detection approach using the notion of visual words by developing a variational framework of finite Dirichlet mixture models. As we shall see clearly from the experimental results, the proposed method is efficient and allows simultaneously the estimation of the parameters of the mixture model and the number of mixture components.

The rest of this paper is organized as follows. In section 2, we present our statistical model. A complete variational approach for its learning is presented

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 276–283, 2011.
© Springer-Verlag Berlin Heidelberg 2011

in section 3. Section 4, is devoted to the experimental results. We end the paper with a conclusion in section 5.

2 Model Specification

The Dirichlet distribution is the multivariate extension of the beta distribution. Define $\boldsymbol{X} = (X_1, ..., X_D)$ as vector of features representing a given object and $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_D)$, where $\sum_{l=1}^{D} X_l = 1$ and $0 \leq X_l \leq 1$ for $l = 1, ..., D$, the Dirichlet distribution is defined as

$$\mathrm{Dir}(\boldsymbol{X}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{l=1}^{D} \alpha_l)}{\prod_{l=1}^{D} \Gamma(\alpha_l)} \prod_{l=1}^{D} X_l^{\alpha_l - 1} \tag{1}$$

where $\Gamma(\cdot)$ is the gamma function defined as $\Gamma(\alpha) = \int_0^{\infty} u^{\alpha-1} e^{-u} du$. Note that in order to ensure the that distribution can be normalized, the constraint $\alpha_l > 0$ must be satisfied. A finite mixture of Dirichlet distributions with M components is represented by [17,18,19]: $p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\alpha}) = \sum_{j=1}^{M} \pi_j \mathrm{Dir}(\boldsymbol{X}|\boldsymbol{\alpha}_j)$, where $\boldsymbol{X} = \{X_1, ..., X_D\}$, $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_M\}$ and $\mathrm{Dir}(\boldsymbol{X}|\boldsymbol{\alpha}_j)$ is the Dirichlet distribution of component j with its own parameters $\boldsymbol{\alpha}_j = \{\alpha_{j1}, ..., \alpha_{jD}\}$. π_j are called mixing coefficients and satisfy the following constraints: $0 \leq \pi_j \leq 1$ and $\sum_{j=1}^{M} \pi_j = 1$. Consider a set of N independent identically distributed vectors $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$ assumed to be generated from the mixture distribution, the likelihood function of the Dirichlet mixture model is given by

$$p(\mathcal{X}|\boldsymbol{\pi}, \boldsymbol{\alpha}) = \prod_{i=1}^{N} \left\{ \sum_{j=1}^{M} \pi_j \mathrm{Dir}(\boldsymbol{X}_i|\boldsymbol{\alpha}_j) \right\} \tag{2}$$

For each vector $\boldsymbol{X}_i$, we introduce a M-dimensional binary random vector $\boldsymbol{Z}_i = \{Z_{i1}, \ldots, Z_{iM}\}$, such that $Z_{ij} \in \{0, 1\}$, $\sum_{j=1}^{M} Z_{ij} = 1$ and $Z_{ij} = 1$ if $\boldsymbol{X}_i$ belongs to component j and 0, otherwise. For the latent variables $\mathcal{Z} = \{\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_N\}$, which are actually hidden variables that do not appear explicitly in the model, the conditional distribution of $\mathcal{Z}$ given the mixing coefficients $\boldsymbol{\pi}$ is defined as $p(\mathcal{Z}|\boldsymbol{\pi}) = \prod_{i=1}^{N} \prod_{j=1}^{M} \pi_j^{Z_{ij}}$. Then, the likelihood function with latent variables, which is actually the conditional distribution of data set $\mathcal{X}$ given the class labels $\mathcal{Z}$ can be written as $p(\mathcal{X}|\mathcal{Z}, \boldsymbol{\alpha}) = \prod_{i=1}^{N} \prod_{j=1}^{M} \mathrm{Dir}(\boldsymbol{X}_i|\boldsymbol{\alpha}_j)^{Z_{ij}}$. In [17], we have proposed an approach based on maximum likelihood estimation for the learning of the finite Dirichlet mixture. However, it has been shown in recent research works that variational learning may provide better results. Thus, we propose in the following a variational approach for our mixture learning.

3 Variational Learning

In this section, we adopt the variational inference methodology proposed in [20] for finite Gaussian mixtures. Inspired from [21], we adopt a Gamma prior:

$\mathcal{G}(\alpha_{jl}|u_{jl}, v_{jl})$ for each α_{jl} to approximate the conjugate prior, where $\boldsymbol{u} = \{u_{jl}\}$ and $\boldsymbol{v} = \{v_{jl}\}$ are hyperparameters, subject to the constraints $u_{jl} > 0$ and $v_{jl} > 0$. Using this prior, we obtain the joint distribution of all the random variables, conditioned on the mixing coefficients:

$$p(\mathcal{X}, \mathcal{Z}, \boldsymbol{\alpha}|\boldsymbol{\pi}) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\pi_j \frac{\Gamma(\sum_{l=1}^{D}\alpha_{jl})}{\prod_{l=1}^{D}\Gamma(\alpha_{jl})}\prod_{l=1}^{D}X_{il}^{\alpha_{jl}-1}\right]^{Z_{ij}} \prod_{j=1}^{M}\prod_{l=1}^{D}\frac{v_{jl}^{u_{jl}}}{\Gamma(u_{jl})}\alpha_{jl}^{u_{jl}-1}e^{-v_{jl}\alpha_{jl}}$$

The goal of the variational learning here is to find a tractable lower bound on $p(\mathcal{X}|\boldsymbol{\pi})$. To simplify the notation without loss of generality, we define $\Theta = \{\mathcal{Z}, \boldsymbol{\alpha}\}$. By applying Jensen's inequality, the lower bound $\mathcal{L}$ of the logarithm of the marginal likelihood $p(\mathcal{X}|\boldsymbol{\pi})$ can be found as

$$\ln p(\mathcal{X}|\boldsymbol{\pi}) = \ln \int Q(\Theta)\frac{p(\mathcal{X}, \Theta|\boldsymbol{\pi})}{Q(\Theta)}d\Theta \geq \int Q(\Theta)\ln\frac{p(\mathcal{X}, \Theta|\boldsymbol{\pi})}{Q(\Theta)}d\Theta = \mathcal{L}(Q) \quad (3)$$

where $Q(\Theta)$ is an approximation to the true posterior distribution $p(\Theta|\mathcal{X}, \boldsymbol{\pi})$. In our work, we adopt the factorial approximation [20,22] for the variational inference. Then, $Q(\Theta)$ can be factorized into disjoint tractable distributions as follows: $Q(\Theta) = Q(\mathcal{Z})Q(\boldsymbol{\alpha})$. In order to maximize the lower bound $\mathcal{L}(Q)$, we need to make a variational optimization of $\mathcal{L}(Q)$ with respect to each of the factors in turn using the general expression for its optimal solution: $Q_s(\Theta_s) = \frac{\exp\left\langle \ln p(\mathcal{X},\Theta)\right\rangle_{\neq s}}{\int \exp\left\langle \ln p(\mathcal{X},\Theta)\right\rangle_{\neq s} d\Theta}$ where $\langle\cdot\rangle_{\neq s}$ denotes an expectation with respect to all the factor distributions except for s. Then, we obtain the optimal solutions as

$$Q(\mathcal{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{M} r_{ij}^{Z_{ij}} \qquad Q(\boldsymbol{\alpha}) = \prod_{j=1}^{M}\prod_{l=1}^{D}\mathcal{G}(\alpha_{jl}|u_{jl}^*, v_{jl}^*) \quad (4)$$

where $r_{ij} = \frac{\rho_{ij}}{\sum_{j=1}^{M}\rho_{ij}}$, $\rho_{ij} = \exp\left\{\ln\pi_j + \widetilde{\mathcal{R}}_j + \sum_{l=1}^{D}(\bar{\alpha}_{jl}-1)\ln X_{il}\right\}$, $u_{jl}^* = u_{jl} + \varphi_{jl}$ and $v_{jl}^* = v_{jl} - \vartheta_{jl}$

$$\begin{aligned}
\widetilde{\mathcal{R}}_j = {} & \ln\frac{\Gamma(\sum_{l=1}^{D}\bar{\alpha}_{jl})}{\prod_{l=1}^{D}\Gamma(\bar{\alpha}_{jl})} + \sum_{l=1}^{D}\bar{\alpha}_{jl}\left[\Psi(\sum_{l=1}^{D}\bar{\alpha}_{jl}) - \Psi(\bar{\alpha}_{jl})\right]\left[\left\langle\ln\alpha_{jl}\right\rangle - \ln\bar{\alpha}_{jl}\right] \\
& + \sum_{l=1}^{D}\bar{\alpha}_{jl}\left[\Psi(\sum_{l=1}^{D}\bar{\alpha}_{jl}) - \Psi(\bar{\alpha}_{jl})\right]\left[\left\langle\ln\alpha_{jl}\right\rangle - \ln\bar{\alpha}_{jl}\right] \\
& + \frac{1}{2}\sum_{l=1}^{D}\bar{\alpha}_{jl}^2\left[\Psi'(\sum_{l=1}^{D}\bar{\alpha}_{jl}) - \Psi'(\bar{\alpha}_{jl})\right]\left\langle(\ln\alpha_{jl} - \ln\bar{\alpha}_{jl})^2\right\rangle \\
& + \frac{1}{2}\sum_{a=1}^{D}\sum_{b=1,a\neq b}^{D}\left\{\Psi'(\sum_{l=1}^{D}\bar{\alpha}_{jl})(\left\langle\ln\alpha_{ja}\right\rangle - \ln\bar{\alpha}_{ja})(\left\langle\ln\alpha_{jb}\right\rangle - \ln\bar{\alpha}_{jb})\right\} \quad (5)
\end{aligned}$$

$$\vartheta_{jl} = \sum_{i=1}^{N} \langle Z_{ij} \rangle \ln X_{il} \tag{6}$$

$$\varphi_{jl} = \sum_{i=1}^{N} \langle Z_{ij} \rangle \bar{\alpha}_{jl} \left[\Psi(\sum_{k=1}^{D} \bar{\alpha}_{jk}) - \Psi(\bar{\alpha}_{jl}) + \sum_{k \neq l}^{D} \Psi'(\sum_{k=1}^{D} \bar{\alpha}_k) \bar{\alpha}_k (\langle \ln \alpha_k \rangle - \ln \bar{\alpha}_k) \right]$$

where $\Psi(\cdot)$ and $\Psi'(\cdot)$ are the digamma and trigamma functions, respectively. The expected values in the above formulas are

$$\langle Z_{ij} \rangle = r_{ij} \ , \qquad \bar{\alpha}_{jl} = \langle \alpha_{jl} \rangle = \frac{u_{jl}}{v_{jl}} \ , \qquad \langle \ln \alpha_{jl} \rangle = \Psi(u_{jl}) - \ln v_{jl}$$
$$\langle (\ln \alpha_{jl} - \ln \bar{\alpha}_{jl})^2 \rangle = [\Psi(u_{jl}) - \ln u_{jl}]^2 + \Psi'(u_{jl})$$

Notice that, $\widetilde{\mathcal{R}}_j$ is the approximate lower bound of $\mathcal{R}_j$, where $\mathcal{R}_j$ is defined as

$$\mathcal{R}_j = \left\langle \ln \frac{\Gamma(\sum_{l=1}^{D} \alpha_{jl})}{\prod_{l=1}^{D} \Gamma(\alpha_{jl})} \right\rangle$$

Unfortunately, a closed-form expression cannot be found for $\mathcal{R}_j$, so the standard variational inference can not be applied directly. Thus, we apply the second-order Taylor series expansion to find a lower bound approximation $\widetilde{\mathcal{R}}_j$ for the variational inference. The solutions to the variational factors $Q(\mathcal{Z})$ and $Q(\boldsymbol{\alpha})$ can be obtained by Eq. 4. Since they are coupled together through the expected values of the other factor, these solutions can be obtained iteratively as discussed above. After obtaining the functional forms for the variational factors $Q(\mathcal{Z})$ and $Q(\boldsymbol{\alpha})$, the lower bound in Eq. 3 of the variational Dirichlet mixture can be evaluated as follows

$$\mathcal{L}(Q) = \sum_{\mathcal{Z}} \int Q(\mathcal{Z}, \boldsymbol{\alpha}) \ln \left\{ \frac{p(\mathcal{X}, \mathcal{Z}, \boldsymbol{\alpha} | \boldsymbol{\pi})}{Q(\mathcal{Z}, \boldsymbol{\alpha})} \right\} d\boldsymbol{\alpha} = \langle \ln p(\mathcal{X}, \mathcal{Z}, \boldsymbol{\alpha} | \boldsymbol{\pi}) \rangle - \langle \ln Q(\mathcal{Z}, \boldsymbol{\alpha}) \rangle$$
$$= \langle \ln p(\mathcal{X} | \mathcal{Z}, \boldsymbol{\alpha}) \rangle + \langle \ln p(\mathcal{Z} | \boldsymbol{\pi}) \rangle + \langle \ln p(\boldsymbol{\alpha}) \rangle - \langle \ln Q(\mathcal{Z}) \rangle - \langle \ln Q(\boldsymbol{\alpha}) \rangle \tag{7}$$

where each expectation is evaluated with respect to all of the random variables in its argument. These expectations are defined as

$$\langle \ln p(\mathcal{X} | \mathcal{Z}, \boldsymbol{\alpha}) \rangle = \sum_{i=1}^{N} \sum_{j=1}^{M} r_{ij} [\widetilde{\mathcal{R}}_j + \sum_{l=1}^{D} (\bar{\alpha}_{jl}) \ln X_{il}]$$

$$\langle \ln p(\mathcal{Z} | \boldsymbol{\pi}) \rangle = \sum_{i=1}^{N} \sum_{j=1}^{M} r_{ij} \ln \pi_j \qquad \langle \ln Q(\mathcal{Z}) \rangle = \sum_{i=1}^{N} \sum_{j=1}^{M} r_{ij} \ln r_{ij}$$

$$\langle \ln p(\boldsymbol{\alpha}) \rangle = \sum_{j=1}^{M} \sum_{l=1}^{D} \left\{ u_{jl} \ln v_{jl} - \ln \Gamma(u_{jl}) + (u_{jl} - 1) \langle \ln \alpha_{jl} \rangle - v_{jl} \bar{\alpha}_{jl} \right\}$$

$$\langle \ln Q(\boldsymbol{\alpha}) \rangle = \sum_{j=1}^{M} \sum_{l=1}^{D} \left\{ u^*_{jl} \ln v^*_{jl} - \ln \Gamma(u^*_{jl}) + (u^*_{jl} - 1) \langle \ln \alpha_{jl} \rangle - v^*_{jl} \bar{\alpha}_{jl} \right\}$$

At each iteration of the re-estimating step, the value of this lower bound should never decrease. The mixing coefficients can be estimated by maximizing the bound $\mathcal{L}(Q)$ with respect to $\boldsymbol{\pi}$. Setting the derivative of this lower bound with respect to $\boldsymbol{\pi}$ to zero gives:

$$\pi_j = \frac{1}{N} \sum_{i=1}^{N} r_{ij} \tag{8}$$

Since the solutions for the variational posterior Q and the value of the lower bound depend on $\boldsymbol{\pi}$, the optimization of the variational Dirichlet mixture model can be solved using an EM-like algorithm with a guaranteed convergence. The complete algorithm can be summarized as follows[1]:

1. Initialization
   - Choose the initial number of components. and the initial values for hyper-parameters $\{u_{jl}\}$ and $\{v_{jl}\}$.
   - Initialize the value of r_{ij} by K-Means algorithm.
2. The variational E-step: Update the variational solutions for $Q(\mathcal{Z})$ and $Q(\boldsymbol{\alpha})$ using Eq. 4.
3. The variational M-step: maximize lower bound $\mathcal{L}(Q)$ with respect to the current value of $\boldsymbol{\pi}$ (Eq. 8).
4. Repeat steps 2 and 3 until convergence (i.e. stabilization of the variational lower bound in (Eq. 7)).
5. Detect the correct M by eliminating the components with small mixing coefficients (less than 10^{-5}).

4 Experimental Results: Object Detection

In this section, we test the performance of the proposed variational Dirichlet mixture (varDM) model on four challenging real-world data sets that have been considered in several research papers in the past for different problems (see, for instance, [7]): Weizmann horse [9], UIUC car [8], Caltech face and Caltech motorbike data sets [2]. Sample images from the different data sets are displayed in Fig. 1. It is noteworthy that the main goal of this section, is to validate our learning algorithm and compare our approach with comparable mixture-based



Horse Car Face Motorbike

Fig. 1. Sample image from each data set

[1] The complete source code is available upon request.
[2] http://www.robots.ox.ac.uk/$\tilde{\text{v}}$gg/data.html.

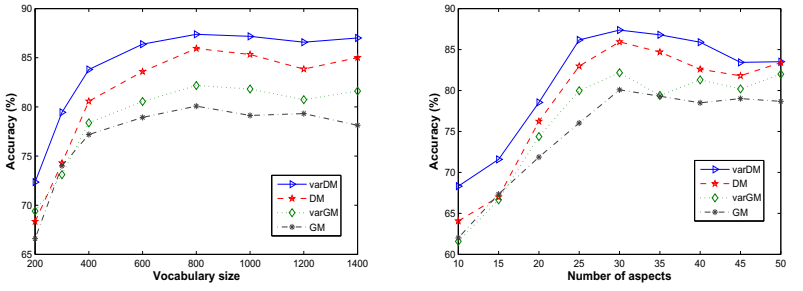
techniques. Thus, comparing with the different object detection techniques that have been proposed in the past is clearly beyond the scope of this paper. We compare the efficiency of our approach with four other approaches for detecting objects in static images: the deterministic Dirichlet mixture model (DM) proposed in [17], the variational Gaussian mixture model (varGM) [20] and the well-known deterministic Gaussian mixture model (GM). In order to provide broad non-informative prior distributions, the initial values of the hyperparameters $\{u_{jl}\}$ and $\{v_{jl}\}$ are set to 1 and 0.01, respectively.

Our methodology for unsupervised object detection can be summarized as follows: First, SIFT descriptors are extracted from each image using the Difference-of-Gaussians (DoG) interest point detectors [23]. Next, a visual vocabulary $\mathcal{W}$ is constructed by quantizing these SIFT vectors into visual words w using K-means algorithm and each image is then represented as the frequency histogram over the visual words. Then, we apply the pLSA model to the bag of visual words representation which allows the description of each image as a D-dimensional vector of proportions where D is the number of learnt topics (or aspects). Finally, we employ our varDM model as a classifier to detect objects by assigning the testing image to the group (object or non-object) which has the highest posterior probability according to Bayes' decision rule.

Each data set is randomly divided into two halves: the training and the testing set considered as positive examples. We evaluated the detection performance of the proposed algorithm by running it 20 times. The experimental results for all the data sets are summarized in Table 1. It clearly shows that our algorithm outperforms the other algorithms for detecting the specified objects. As expected, we notice that varGM and GM perform worse than varDM and DM. Since compared to Gaussian mixture model, recent works have shown that Dirichlet mixture model may provide better modeling capabilities in the case of non-Gaussian data in general and proportional data in particular [24]. We have also tested the effect of different sizes of visual vocabulary on detection accuracy for varDM, DM, varGM and GM, as illustrated in Fig. 2(a). As we can see, the detection rate peaks around 800. The choice of the number of aspects also influences the accuracy of detection. As shown in Fig. 2(b), the optimal accuracy can be obtained when the number of aspects is set to 30.

Table 1. The detection rate (%) on different data set using different approaches

	varDM	DM	varGM	GM
Horse	87.38	85.94	82.17	80.08
Car	84.83	83.06	80.51	78.13
Face	88.56	86.43	82.24	79.38
Motorbike	90.18	86.65	85.49	81.21



(a) (b)

Fig. 2. (a) Detection accuracy vs. the number of aspects for the horse data set; (b) Feature saliencies for the different aspect features over 20 runs for the horse data set

5 Conclusion

In our work, we have proposed a variational framework for finite Dirichlet mixture models. By applying the varDM model with pLSA, we built an unsupervised learning approach for object detection. Experimental results have shown that our approach is able to successfully and efficiently detect specific objects in static images. The proposed approach can be applied also to many other problems which involve proportional data modeling and clustering such as text mining, analysis of gene expression data and natural language processing. A promising future work could be the extension of this work to the infinite case as done in [25].

Acknowledgment. The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Papageorgiou, C.P., Oren, M., Poggio, T.: A General Framework for Object Detection. In: Proc. of ICCV, pp. 555–562 (1998)
2. Viitaniemi, V., Laaksonen, J.: Techniques for Still Image Scene Classification and Object Detection. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4132, pp. 35–44. Springer, Heidelberg (2006)
3. Chen, H.F., Belhumeur, P.N., Jacobs, D.W.: In Search of Illumination Invariants. In: Proc. of CVPR, pp. 254–261 (2000)
4. Cootes, T.F., Walker, K., Taylor, C.J.: View-Based Active Appearance Models. In: Proc. of FGR, pp. 227–232 (2000)
5. Gross, R., Matthews, I., Baker, S.: Eigen Light-Fields and Face Recognition Across Pose. In: Proc. of FGR, pp. 1–7 (2002)
6. Rowley, H.A., Baluja, S., Kanade, T.: Human Face Detection in Visual Scenes. In: Proc. of NIPS, pp. 875–881 (1995)
7. Shotton, J., Blake, A., Cipolla, R.: Contour-Based Learning for Object Detection. In: Proc. of ICCV, pp. 503–510 (2005)

8. Agarwal, S., Roth, D.: Learning a Sparse Representation for Object Detection. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part IV. LNCS, vol. 2353, pp. 113–127. Springer, Heidelberg (2002)
9. Borenstein, E., Ullman, S.: Learning to segment. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004, Part III. LNCS, vol. 3023, pp. 315–328. Springer, Heidelberg (2004)
10. Papageorgiou, C., Poggio, T.: A Trainable System for Object Detection. International Journal of Computer Vision 38(1), 15–23 (2000)
11. Fergus, R., Perona, P., Zisserman, A.: Object Class Recognition by Unsupervised Scale-Invariant Learning. In: Proc. of CVPR, pp. 264–271 (2003)
12. Bosch, A., Zisserman, A., Muñoz, X.: Scene Classification via pLSA. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part IV. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)
13. Boutemedjet, S., Bouguila, N., Ziou, D.: A Hybrid Feature Extraction Selection Approach for High-Dimensional Non-Gaussian Data Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(8), 1429–1443 (2009)
14. Boutemedjet, S., Ziou, D., Bouguila, N.: Unsupervised Feature Selection for Accurate Recommendation of High-Dimensional Image Data. In: NIPS, pp. 177–184 (2007)
15. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
16. Hofmann, T.: Probabilistic Latent Semantic Indexing. In: Proc. of ACM SIGIR, pp. 50–57 (1999)
17. Bouguila, N., Ziou, D., Vaillancourt, J.: Unsupervised Learning of a Finite Mixture Model Based on the Dirichlet Distribution and Its Application. IEEE Transactions on Image Processing 13(11), 1533–1543 (2004)
18. Bouguila, N., Ziou, D.: Using unsupervised learning of a finite Dirichlet mixture model to improve pattern recognition applications. Pattern Recognition Letters 26(12), 1916–1925 (2005)
19. Bouguila, N., Ziou, D.: Online Clustering via Finite Mixtures of Dirichlet and Minimum Message Length. Engineering Applications of Artificial Intelligence 19(4), 371–379 (2006)
20. Corduneanu, A., Bishop, C.M.: Variational Bayesian Model Selection for Mixture Distributions. In: Proc. of AISTAT, pp. 27–34 (2001)
21. Ma, Z., Leijon, A.: Bayesian Estimation of Beta Mixture Models with Variational Inference. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (2010) (in press)
22. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. In: Learning in Graphical Models, pp. 105–162. Kluwer (1998)
23. Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. IEEE TPAMI 27(10), 1615–1630 (2005)
24. Bouguila, N., Ziou, D.: Unsupervised Selection of a Finite Dirichlet Mixture Model: An MML-Based Approach. IEEE Transactions on Knowledge and Data Eng. 18(8), 993–1009 (2006)
25. Bouguila, N., Ziou, D.: A Dirichlet Process Mixture of Dirichlet Distributions for Classification and Prediction. In: Proc. of the IEEE Workshop on Machine Learning for Signal Processing (MLSP), pp. 297–302 (2008)

Performances Evaluation of GMM-UBM and GMM-SVM for Speaker Recognition in Realistic World

Nassim Asbai, Abderrahmane Amrouche, and Mohamed Debyeche

Speech Communication and Signal Processing Laboratory,
Faculty of Electronics and Computer Sciences, USTHB,
P.O. Box 32, El Alia, Bab Ezzouar, 16111, Algiers, Algeria
{asbainassim,mdebyeche}@gmail.com, namrouche@usthb.dz

Abstract. In this paper, an automatic speaker recognition system for realistic environments is presented. In fact, most of the existing speaker recognition methods, which have shown to be highly efficient under noise free conditions, fail drastically in noisy environments. In this work, features vectors, constituted by the Mel Frequency Cepstral Coefficients (MFCC) extracted from the speech signal are used to train the Support Vector Machines (SVM) and Gaussian mixture model (GMM). To reduce the effect of noisy environments the cepstral mean subtraction (CMS) are applied on the MFCC. For both, GMM-UBM and GMM-SVM systems, 2048-mixture UBM is used. The recognition phase was tested with Arabic speakers at different Signal-to-Noise Ratio (SNR) and under three noisy conditions issued from NOISEX-92 data base. The experimental results showed that the use of appropriate kernel functions with SVM improved the global performance of the speaker recognition in noisy environments.

Keywords: Speaker recognition, Noisy environment, MFCC, GMM-UBM, GMM-SVM.

1 Introduction

Automatic speaker recognition (ASR) has been the subject of extensive research over the past few decades [1]. These can be attributed to the growing need for enhanced security in remote identity identification or verification in such applications as telebanking and online access to secure websites. Gaussian Mixture Model (GMM) was the state of the art of speaker recognition techniques [2]. The last years have witnessed the introduction of an effective alternative speaker classification approach based on the use of Support Vector Machines (SVM) [3]. The basis of the approach is that of combining the discriminative characteristics of SVMs [3],[4] with the efficient and effective speaker representation offered by GMM-UBM [5],[6] to obtain hybrid GMM-SVM system [7],[8].

The focus of this paper is to investigate into the effectiveness of the speaker recognition techniques under various mismatched noise conditions. The issue of the Arabic language, customary in more than 300 million peoples around the

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 284–291, 2011.
© Springer-Verlag Berlin Heidelberg 2011

world, which remains poorly endowed in language technologies, challenges us and dictates the choice of a corpus study in this work.

The remainder of the paper is structured as follows. In sections 2 and 3, we discuss the GMM and SVM classification methods and briefly describe the principles of GMM-UBM at section 4. In section 5, experimental results of the speaker recognition in noisy environment using GMM, SVM and GMM-SVM systems based using ARADIGITS corpora are presented. Finally, a conclusion is given in Section 6.

2 Gaussian Mixture Model (GMM)

In GMM model [9], there exist k underlying components $\{\omega_1, \omega_2, ..., \omega_k\}$ in a d-dimensional data set. Each component follows some Gaussian distribution in the space. The parameters of the component ω_j include

$\lambda_j = \{\mu_j, \Sigma_1, ..., \pi_j\}$, in which $\mu_j = (\mu_j[1], ..., \mu_j[d])$ is the center of the Gaussian distribution, $\Sigma_j$is the covariance matrix of the distribution and π_j is the probability of the component ω_j . Based on the parameters, the probability of a point coming from component ω_j appearing at $x_j = x[1], ..., x[d]$ can be represented by

$$\Pr(x/\lambda_j) = \frac{1}{(2\pi)^{d/2}|\sum_j^{-1}|} \exp\{-\frac{1}{2}(x-\mu_j)^T \sum^{-1}(x-\mu_j)\} \tag{1}$$

Thus, given the component parameter set $\{\lambda_1, \lambda_2, ..., \lambda_k\}$ but without any component information on an observation point , the probability of observing is estimated by

$$\Pr(x/\lambda_j) = \sum_{j=1}^{k} Pr(x/\lambda_j)\pi_j \tag{2}$$

The problem of learning GMM is estimating the parameter set λ of the k component to maximize the likelihood of a set of observations $D = \{x_1, x_2, ..., x_n\}$, which is represented by

$$\Pr(D/\lambda) = \Pi_{i=1}^{n} Pr(x_i/\lambda) \tag{3}$$

3 Support Vector Machines (SVM)

SVM is a binary classifier which models the decision boundary between two classes as a separating hyperplane. In speaker verification, one class consists of the target speaker training vectors (labeled as +1), and the other class consists of the training vectors from an "impostor" (background) population (labeled as -1). Using the labeled training vectors, SVM optimizer finds a separating hyperplane that maximizes the margin of separation between these two classes. Formally, the discriminate function of SVM is given by [4]:

$$f(x) = class(x) = sign[\sum_{i=1}^{N} \alpha_i t_i K(x, x_i) + d] \tag{4}$$

Here $t_i \varepsilon \{+1, -1\}$ are the ideal output values, $\sum_{i=1}^{N} \alpha_i t_i = 0$ and $\alpha_i > 0$ ¿ 0. The support vectors x_i, their corresponding weights α_i and the bias term d, are determined from a training set using an optimization process. The kernel function $K(,)$ is designed so that it can be expressed as $K(x, y) = \Phi(x)^T \Phi(y)$ where $\Phi(x)$ is a mapping from the input space to kernel feature space of high dimensionality. The kernel function allows computing inner products of two vectors in the kernel feature space. In a high-dimensional space, the two classes are easier to separate with a hyperplane. To calculate the classification function class (x) we use the dot product in feature space that can also be expressed in the input space by the kernel [13]. Among the most widely used cores we find:

- Linear kernel: $K(u, v) = u.v$;
- Polynomial kernel: $K(u, v) = [(u.v) + 1]^d$;
- RBF kernel: $K(u, v) = \exp(-\gamma |u.v|^2)$.

SVMs were originally designed primarily for binary classification [11]. Their extension problem of multi-class classification is still a research topic. This problem is solved by combining several binary SVMs.

One against all: This method constructs K SVMs models (one SVM for each class). The ith SVM is learned with all the examples. The ith class is indexed with positive labels and all others with negative labels. This ith classifier builds hyperplane between the ith class and other K -1 class.

One against one: This method constructs $K(K-1)/2$ classifiers where each is learned on data from two classes. During the test phase and after construction of all classifiers, we use the proposed voting strategy.

4 GMM-UBM and GMM-SVM Systems

The GMM-UBM [2] system implemented for the purpose of this study uses MAP [12] estimation to adapt the parameters of each speaker GMM from a clean gender balanced UBM. For the purpose of consistency, a 2048-mixture UBM is used for both GMM-UBM and GMM-SVM systems. In the GMM-SVM system, the GMMs are obtained from training, testing and background utterances using the same procedure as that in the GMM-UBM system. Each client training supervector is assigned a label of +1 whereas the set of supervectors from a background dataset representing a large number of impostors is given a label of -1.The procedure used for extracting supervectors in the testing phase is exactly the same as that in the training stage (in the testing phase, no labels are given to the supervectors).

5 Results and Discussion

5.1 Experimental Protocol and Data Collection

Arabic digits, which are polysyllabic, can be considered as representative elements of language, because more than half of the phonemes of the Arabic language are included in the ten digits. The speech database used in this work is

a part of the database ARADIGITS [13]. It consists of a set of 10 digits of the Arabic language (zero to nine) spoken by 60 speakers of both genders with three repetitions for each digit. This database was recorded by speakers from different regions Algerians aged between 18 and 50 years in a quiet environment with an ambient noise level below 35 dB, in WAV format, with a sampling frequency equal to 16 kHz. To simulate the real environment we used noises extracted from the database Noisex-92 (NATO: AC 243/RSG 10). In parameterization phase, we specified the feature space used. Indeed, as the speech signal is dynamic and variable, we presented the observation sequences of various sizes by vectors of fixed size. Each vector is given by the concatenation of the coefficients mel cepstrum MFCC (12 coefficients), these first and second derivatives (24 coefficients), extracted from the middle window every 10 ms. A cepstral mean subtraction (CMS) is applied to these features in order to reduce the effect of noise.

5.2 Speaker Recognition in Quiet Environment Using GMM and SVM

The experimental results, given in Fig.1, show that the performances are better for males speakers (98, 33%) than females (96, 88%). The recognition rate is better for a GMM with k = 32 components (98.19%) than other GMMs with other numbers of components. Now, if we compare between the performances of classifiers (GMM and SVM), we note that GMM with k = 32 components yields better results than SVM (linear SVM (88.33%), SVM with RBF kernel (86.36%) and SVM with polynomial kernel with degree d = 2 (82.78%)).

5.3 Speaker Recognition in Noisy Environments Using GMM and SVM

In this part we add noises (of factory and military engine) extracted from the NATO base NOISEX'92 (Varga), to our test database ARADIGITS that

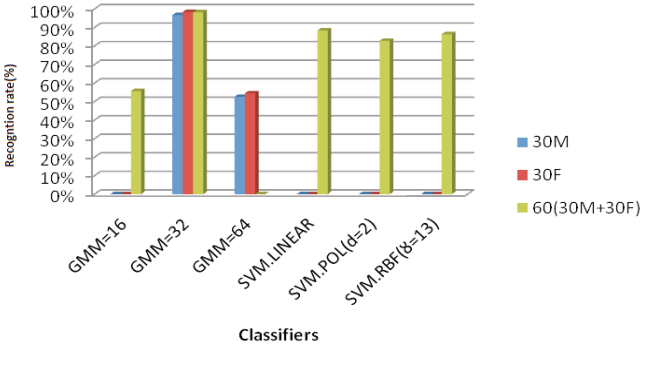


Fig. 1. Histograms of the recognition rate of different classifiers used in a quiet environment

containing 60 speakers (30 male and 30 female). From the results presented in Fig.2 and Fig.3, we find that the SVMs are more robust than the GMM. For example, recognition rate equal to 67.5%.(for SVN using polynomial kernel with d=2). than GMM used in this work. But, in other noise (factory noise) we find that GMM (with k=32) gives better performances (recognition rate equal to 61.5% with noise of factory at SNR = 0dB) than SVM. This implies that SVMs and GMM (k=32) are more suitable for speaker recognition in a noisy environment and also we note that the recognition rate varies from noise to another. As that as far as the SNR increases (less noise), recognition is better.

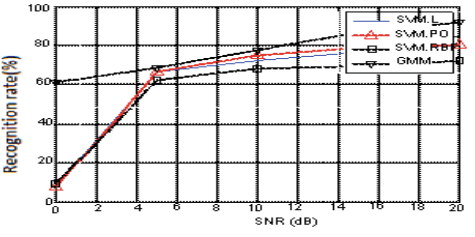


Fig. 2. Performances evaluation for speaker recognition systems in noisy environment corrupted by noise of factory

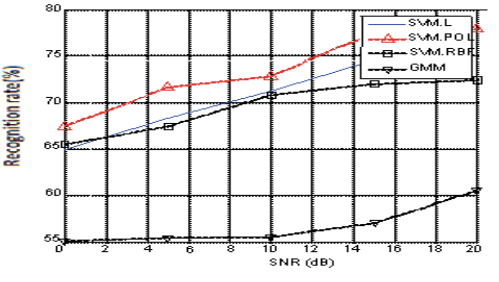


Fig. 3. Performances evaluation for speaker recognition systems in noisy environment corrupted by military engine

5.4 Speaker Recognition in Quiet Environment Using GMM-UBM and GMM-SVM

The result in terms of equal-error rate (EER) shown by DET curve (Detection Error trade-off curve) showed in Fig.4:

1. When the GMM supervector is used, with MAP estimation [12], as input to the SVMs, the EER is 2.10%.
2. When the GMM-UBM is used the EER is 1.66%.

In the quiet environment, we can say that, the performances of GMM-UBM and GMM-SVM are almost similar with a slight advantage for GMM-UBM.

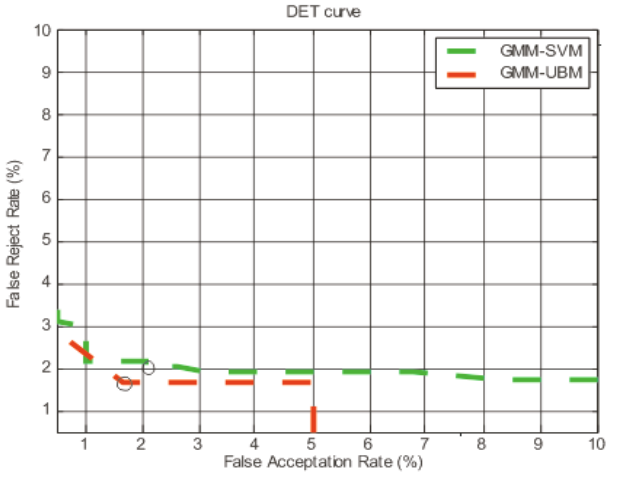


Fig. 4. DET curve for GMM-UBM and GMM-SVM

5.5 Speaker Recognition in Noisy Environments Using GMM-UBM and GMM-SVM

The goal of the experiments doing in this section is to evaluate the recognition performances of GMM-UBM and GMM-SVM when the quality of the speech data is contaminated with different levels of different noises extracted from the NOISEX'92 database. This provides a range of speech SNRs (0, 5, and 10 dB). Table 1 and 2 present the experimental results in terms of equal error rate (EER) in real world. As expected, it is seen that there is a drop in accuracy for this approaches with decreasing SNR.

Table 1. EER in speaker recognition experiments with GMM-UBM method under mismatched data condition using different noises

GMM-UBM						
Noise	Test data					
	SNR : 0 dB		SNR : 5 dB		SNR : 10 dB	
	EER %	MinDCF	EER %	MinDCF	EER %	MinDCF
Leopard	15	0.053	6.66	0.020	3.33	0.018
Factory	9.33	0.093	7.33	0.083	1.67	0.021
Babble-speech	10	0.023	5	0.023	3.33	0.018

The experimental results given in Table 1 and 2 show that the EERs for GMM-SVM are higher for mismatched conditions noise. We can observe that, the difference between EERs in clean and noisy environment for two systems GMM-UBM and GMM-SVM. So, it is noted that again, the usefulness of GMM-SVM in reducing error rates is noisy environment against GMM-UBM.

Table 2. EERs in speaker recognition experiments with GMM-SVM method under mismatched data condition using different noises

GMM-SVM						
Noise	Test data					
	SNR : 0 dB		SNR : 5 dB		SNR : 10 dB	
	EER %	MinDCF	EER %	MinDCF	EER %	MinDCF
Leopard	2.60	0.0084	2.45	0.0081	2.16	0.0080
Factory	3.23	0.0079	2.56	0.0079	2.24	0.0078
Babble-speech	2.85	0.0084	2.37	0.0083	2.13	0.0081

6 Conclusion

The aim of our study in this paper was to evaluate the contribution of kernel methods in improving system performance of automatic speaker recognition (RAL) (identification and verification) in the real environment, often represented by an acoustic environment highly degraded. Indeed, the determination of physical characteristics discriminating one speaker from another is a very difficult task, especially in adverse environment. For this, we developed a system of automatic speaker recognition on text independent mode, part of which recognition is based on classifier using kernel functions, which are alternatively SVM (with linear, polynomial and radial kernels) and GMM.

On the other hand, we used GMM-UBM, especially the system hybrid GMM-SVM, which the vector means extracted from GMM-UBM with 2048 mixtures for UBM in step of modeling are inputs for SVMs in phase of decision.

The results we have achieved conform all that SVM and SVM-GMM techniques are very interesting and promising especially for tasks such as recognition in a noisy environments.

References

1. Dong, X., Zhaohui, W.: Speaker Recognition using Continuous Density Support Vector Machines. Electronics Letters 37, 1099–1101 (2001)
2. Reynolds, D.A., Quatiery, T., Dunn, R.: Speaker Verification Using Adapted Gaussian Mixture Models. Dig. Signal Process. 10, 19–41 (2000)
3. Cristianni, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press (2000)
4. Wan, V.: Speaker Verification Using Support Vector Machines, Ph.D Thesis, University of Sheffield (2003)
5. Campbel, W.M., Sturim, D.E., Reynolds, D.A.: Support Vector Machines Using GMM Supervectors for Speaker Verification. IEEE Signal Process. Lett. 13(5), 115–118 (2006)
6. Minghui, L., Yanlu, X., Zhigiang, Y., Beigian, D.: A New Hybrid GMM/SVM for Speaker Verification. In: Proc. Int. Conf. Pattern Recognition, vol. 4, pp. 314–317 (2006)

7. Campbel, W.M., Sturim, D.E., Reynolds, D.A., Solomonoff, A.: SVM Based Speaker Verification Using a GMM Supervector Kernel and NAP Variability Compensation. In: Proc. IEEE Conf. Acoustics, Speech and Signal Processing, vol. 1, pp. 97–100 (2007)
8. Dehak, R., Dehak, N., Kenny, P., Dumouchel, P.: Linear and Non Linear Kernel GMM Supervector Machines for Speaker Verification. In: Proc. Interspeech, pp. 302–305 (2007)
9. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley-Interscience (2000)
10. Moreno, P.J., Ho, P.P., Vasconcelos, N.: A Generative Model Based Kernel for SVM Classification in Multimedia Applications. In: Neural Informations Processing Systems (2003)
11. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning 20(3), 273–297 (1995)
12. Ben, M., Bimbot, F.: D-MAP: a Distance-Normalized MAP Estimation of Speaker Models for Automatic Speaker Verification. In: Proc. IEEE Conf. Acoustics, Speech and Signal Processing, vol. 2, pp. 69–72 (2008)
13. Amrouche, A., Debyeche, M., Taleb Ahmed, A., Rouvaen, J.M., Ygoub, M.C.E.: Efficient System for Speech Recognition in Adverse Conditions Using Nonparametric Regression. Engineering Applications on Artificial Intelligence 23(1), 85–94 (2010)

SVM and Greedy GMM Applied on Target Identification

Dalila Yessad, Abderrahmane Amrouche, and Mohamed Debyeche

Speech Communication and Signal Processing Laboratory,
Faculty of Electronics and Computer Sciences, USTHB,
P.O. Box 32, El Alia, Bab Ezzouar, 16111, Algiers, Algeria
{yessad.dalila,mdebyeche}@gmail.com, namrouche@usthb.dz

Abstract. This paper is focused on the Automatic Target Recognition (ATR) using Support Vector Machines (SVM) combined with automatic speech recognition (ASR) techniques. The problem of performing recognition can be broken into three stages: data acquisition, feature extraction and classification. In this work, extracted features from micro-Doppler echoes signal, using MFCC, LPCC and LPC, are used to estimate models for target classification. In classification stage, three parametric models based on SVM, Gaussian Mixture Model (GMM) and Greedy GMM were successively investigated for echo target modeling. Maximum a posteriori (MAP) and Majority-voting post-processing (MV) decision schemes are applied. Thus, ASR techniques based on SVM, GMM and GMM Greedy classifiers have been successfully used to distinguish different classes of targets echoes (humans, truck, vehicle and clutter) recorded by a low-resolution ground surveillance Doppler radar. The obtained performances show a high rate correct classification on the testing set.

Keywords: Automatic Target Recognition (ATR), Mel Frequency Cepstrum Coefficients (MFCC), Support Vector Machines (SVM), Greedy Gaussian Mixture Model (Greedy GMM), Majority Vot processing (MV).

1 Introduction

The goal for any target recognition system is to give the most accurate interpretation of what a target is at any given point in time. Techniques based on [1] Micro-Doppler signatures [1, 2] are used to divide targets into several macro groups such as aircrafts, vehicles, creatures, etc. An effective tool to extract information from this signature is the time-frequency transform [3]. The time-varying trajectories of the different micro-Doppler components are quite revealing, especially when viewed in the joint time-frequency space [4, 5]. Anderson [6] used micro-Doppler features to distinguish among humans, animals and vehicles. In [7], analysis of radar micro-Doppler signature with time-frequency transform, the micro-Doppler phenomenon induced by mechanical vibrations or rotations of structures in a radar target are discussed, The time-frequency signature of the

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 292–299, 2011.
© Springer-Verlag Berlin Heidelberg 2011

micro-Doppler provides additional time information and shows micro-Doppler frequency variations with time. Thus, additional information about vibration rate or rotation rate is available for target recognition. Gaussian mixture model (GMM)-based classification methods are widely applied to speech and speaker recognition [8, 9]. Mixture models form a common technique for probability density estimation. In [8] it was proved that any density can be estimated to a given degree of approximation, using finite Gaussian mixture. A Greedy learning of Gaussian mixture model (GMM) based on target classification for ground surveillance Doppler radar, recently proposed in [9], overcomes the drawbacks of the EM algorithm. The greedy learning algorithm does not require prior knowledge of the number of components in the mixture, because it inherently estimates the model order.

In this paper, we investigate the micro-Doppler radar signatures using three classifiers; SVM, GMM and Greedy GMM. The paper is organized as follows: in section 2, the SVM and Greedy GMM and the corresponding classification scheme are presented. In Section 3, we describe the experimental framework including the data collection of different targets from a ground surveillance radar records and the conducted performance study. Our conclusions are drawn in section 5.

2 Classification Scheme

2.1 Feature Extraction

In practical case, a human operator listen to the audio Doppler output from the surveillance radar for detecting and may be identifying targets. In fact, human operators classify the targets using an audio representation of the micro-Doppler effect, caused by the target motion.

As in speech processing a set of operations are taken during pre-processing step to take in count the human ear characteristics. Features are numerical measurements used in computation to discriminate between classes. In this work, we investigated three classes of features namely, LPC (Linear prediction coding), LPCC (Linear cepstral prediction coding), and MFCC (Mel-frequency cepstral coefficients).

2.2 Modelisation

Gaussian Mixture Model (GMM). Gaussian mixture model (GMM) is a mixture of several Gaussian distributions. The probability density function is defined as a weighted sum of Gaussians:

$$p\left(x;\theta\right) = \sum_{c=1}^{C} \alpha_c N(x;\mu_c,\Sigma_c) \tag{1}$$

Where α_c is the weight of the component c, $0 < \alpha_c < 1$ for all components, and $\sum_{c+1}^{C} \alpha_c = 1$. μ_c is the mean of components and Σ_c is the covariance matrix.

We define the parameter vector θ:

$$\theta = \{\alpha_1, \mu_1, \Sigma_1, ..., \alpha_c, \mu_c, \Sigma_c\} \tag{2}$$

The expectation maximization (EM) algorithm is an iterative method for calculating maximum likelihood distribution parameter. An elegant solution for the initialization problem is provided by the greedy learning of GMM [11].

Greedy Gaussian Mixture Model (Greedy GMM). The greedy algorithm starts with a single component and then adds components into the mixture one by one. The optimal starting component for a Gaussian mixture is trivially computed, optimal meaning the highest training data likelihood. The algorithm repeats two steps: insert a component into the mixture, and run EM until convergence. Inserting a component that increases the likelihood the most is thought to be an easier problem than initializing a whole near-optimal distribution. Component insertion involves searching for the parameters for only one component at a time. Recall that EM finds a local optimum for the distribution parameters, not necessarily the global optimum which makes it initialization dependent method. Let p_c denote a C-component mixture with parameters θ_c. The general greedy algorithm for Gaussian mixture is as follows:

1. Compute (in the ML sense) the optimal one-component mixture p_1 and set $C \leftarrow 1$;
2. While keeping p_c fixed, find a new component $N(x; \mu', \Sigma')$ and the corresponding mixing weight α' that increase the likelihood

$$\{\mu', \Sigma', \alpha'\} = \arg\max \sum_{n=1}^{N} \ln[(1-\alpha)p_c(x_n) + \alpha N(x_n; \mu, \Sigma)] \tag{3}$$

3. Set $p_{c+1}(x) \leftarrow (1-\alpha')p_c(x) + \alpha' N(x; \mu', \Sigma')$ and then $C \leftarrow C + 1$;
4. Update p_c using EM (or some other method) until convergence;
5. Evaluate some stopping criterion; go to step 2 or quit. The stopping criterion in step 5 can be for example any kind of model selection criterion or wanted number of components. The crucial point is step 2, since finding the optimal new component requires a global search, performed by creating candidate components. The candidate resulting in the highest likelihood when inserted into the (previous) mixture is selected. The parameters and weight of the best candidate are then used in step 3 instead of the truly optimal values [12].

2.3 Support Vector Machine (SVM)

The optimization criterion here is the width of the margin between classes (see Fig.1), i.e. the empty area around the decision boundary defined by the distance to the nearest training pattern [13]. These patterns, called support vectors, finally define the classification. Maximizing the margin minimizes the number of support vectors. This can be illustrated in Fig.1 where m is maximized.

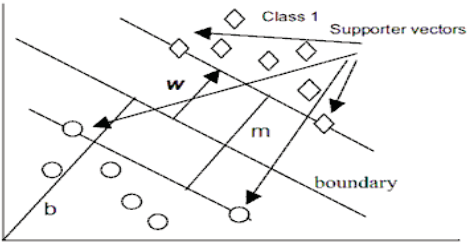


Fig. 1. SVM boundary (It should be as far away from the data of both class as possible)

The general form of the decision boundary is as follows:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i x w + b \tag{4}$$

where α is the Lagrangian coefficient; y is the classes $(+1 or -1)$; w and b are illustrated in Fig.1.

2.4 Classification

A classifier is a function that defines the decision boundary between different patterns (classes). Each classifier must be trained with a training dataset before being used to recognize new patterns, such that it generalizes training dataset into classification rules. Two decision methods were examined. The first one suggests the maximum a posteriori probability (MAP) and the second uses the majority vote (MV) post-processing after classifier decision.

Decision. If we have a group of targets represented by the GMM or SVM models: $\lambda_1, \lambda_2, ..., \lambda_\xi$, The classification decision is done using the posteriori probability (MAP):

$$\hat{S} = \arg\max p(\lambda_s|X) \tag{5}$$

According to Bayesian rule:

$$\hat{S} = \arg\max \frac{p(X|\lambda_s)p(\lambda_s)}{p(X)} \tag{6}$$

X: is the observed sequence.

Assuming that each class has the same a priori probability ($p(\lambda_s) = 1/\xi$) and the probability of apparition of the sequence X is the same for all targets the classification rule of Bayes becomes:

$$\hat{S} = \arg\max p(X|\lambda_s) \tag{7}$$

Majority Vote. The majority vote (MV) post-processing can be employed after classifier decision. It uses the current classification result, along with the previous classification results and makes a classification decision based on the class that appears most often. A plot of the classification by MV (post-processing) after classifier decision is shown in Fig.2.

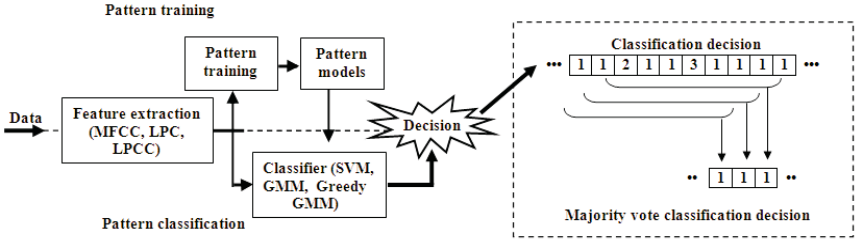


Fig. 2. Majority vote post-processing after classifier decision

3 Radar System and Data Collection

Data were obtained using records of a low-resolution ground surveillance radar. The target was detected and tracked automatically by the radar, allowing continuous target echo records. The parameters settings are: Frequency: 9.720 GHz, Sweep in azimuth: 30 at 270, Emission power : 100 mW.

We first collected the Doppler signatures from the echoes of six different targets in movements namely: one, two, and three persons, vehicle, truck and vegetation clutter. the target was detected and tracked automatically by a low-power Doppler radar operating at 9.72 GHz. When the radar transmits an electromagnetic signal in the surveillance area, this signal interacts with the target and then returns to the radar. After demodulation and analog to digital conversion, the received echoes are recorded in wav audio format, each record has a duration of 10 seconds. By taking the Fourier transform of the recorded signal, the micro-Doppler frequency shift may be observed in the frequency domain. We considered the case where a target approaches the radar. In order to exploit the time-varying Doppler information, we use the short-time Fourier transform (STFT) for the joint MFCC analysis.

The change of the properties of the returned signal reflects the characteristics of the target. When the target is moving, the carrier frequency of the returned signal will be shifted due to Doppler effect. The Doppler frequency shift can be used to determine the radial velocity of the moving target. If the target or any structure on the target is vibrating or rotating in addition to target translation, it will induce frequency modulation on the returned signal that generates sidebands about the target's Doppler frequency. This modulation is called the micro-Doppler (μ-DS) phenomenon. The (μ-DS) phenomenon can be regarded as a characteristic of the interaction between the vibrating or rotating structures and the target body. Fig.3 show the temporal representation and the

typical spectrogram of truck target. The truck class has unique time-frequency characteristic which can be used for classification. This particular plot is obtained by taking a succession of FFTs and using a sampling rate of 8 KHz, FFT size of 256 points, overlap of 128, and a Hamming window.

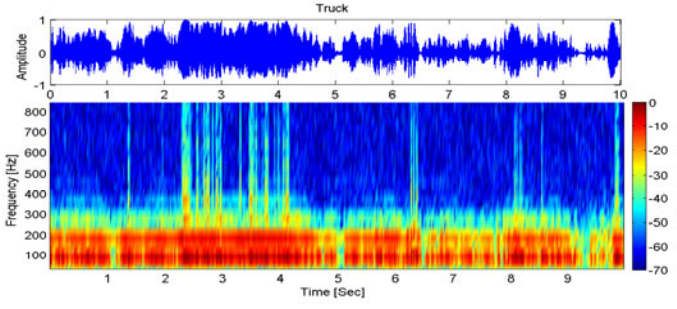


Fig. 3. Radar echos sample (temporal form) and typical spectrogram of the truck moving target

4 Results

In this work, target class pdfs were modeled by SVM and GMMs using both greedy and EM estimation algorithms. MFCC, LPCC and LPC coefficients were used as classification features. The MAP and the majority voting decision concepts were examined. Classification performance obtained using GMM classifier is bad then both GMM greedy and SVM. Table 1 present the confusion matrix of six targets, when the coefficients are extracted MFCC, then classified by GMM following MAP decision and MV post-processing decision. Table 2 show the confusion matrix of six targets classified by SVM following MAP and MV post-processing decision, using MFCC. Table 3 present the confusion matrix of Greedy GMM based classifier with MFCC coefficients and MV post-processing after MAP decision for six class problem. Greedy GMM and SVM outperform GMM classifier. These tables show that both SVM and greedy GMM classifier with MFCC features outperform the GMM based one. To improve classification

Table 1. Confusion matrix of GMM-based classifier with MFCC coefficients and MV post-processing after MAP decision rules for six class problem

Class/Decision	1Person	2Persons	3Persons	Vehicle	Truck	Clutter
1Person	94.44	1.85	0	3.7	0	0
2Persons	0	100	0	0	0	0
3Persons	7.41	0	92.59	0	0	0
Vehicle	12.96	0	0	87.04	0	0
Truck	0	0	0	1.85	98.15	0
Clutter	0	0	0	0	0	100

Table 2. Confusion matrix of SVM-based classifier with MFCC coefficients and MV post-processing after MAP decision rules for six class problem

Class/Decision	1Person	2Persons	3Persons	Vehicle	Truck	Clutter
1Person	96.30	1.85	0	1.85	0	0
2Persons	0	99.07	0.3	0	0	0
3Persons	0	0	100	0	0	0
Vehicle	1.85	0	0	98.15	0	0
Truck	0	0	0	0	100	0
Clutter	0	0	0	0	0	100

Table 3. Confusion matrix of Greedy GMM-based classifier with MFCC coefficients and MV post-processing after MAP decision rules for six class problem

Class/Decision	1Person	2Persons	3Persons	Vehicle	Truck	Clutter
1Person	96.30	1.85	0	1.85	0	0
2Persons	0	100	0	0	0	0
3Persons	0	0	100	0	0	0
Vehicle	1.85	0	0	98.15	0	0
Truck	0	0	0	0	100	0
Clutter	0	0	0	0	0	100

accuracy, majority vote post-processing can be employed. The resulting effect is a smooth operation that removes spurious misclassification. Indeed, the classification rate improves to 99.08% for greedy GMM after MAP decision following majority vote post-processing, 98.93% for GMM and 99.01% for SVM after MAP and MV decision. One can see that the pattern recognition algorithm is quite successful at classifying the radar targets.

5 Conclusion

Automatic classifiers have been successfully applied for ground surveillance radar. LPC, LPCC and MFCC are used to exploit the micro-Doppler signatures of the targets to provide classification between the classes of personnel, vehicle, truck and clutter, The MAP and the majority voting decision rules were applied to the proposed classification problem. We can say that both SVM and Greedy GMM using MFCC features delivers the best rate of classification, as it performs the most estimations. However, it fails to avoid classification errors, which we are bound to eradicate through MV-post processing which guarantees a 99.08% with Greedy GMM and 99.01%withe SVM classification rate for six-class problem in our case.

References

1. Natecz, M., Rytel-Andrianik, R., Wojtkiewicz, A.: Micro-Doppler Analysis of Signal Received by FMCW Radar. In: International Radar Symposium, Germany (2003)

2. Boashash, B.: Time Frequency Signal Analysis and Processing a comprehensive reference, 1st edn. Elsevier Ltd. (2003)
3. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
4. Chen, V.C.: Analysis of Radar Micro-Doppler Signature With Time-Frequency Transform. In: Proc. Tenth IEEE Workshop on Statistical Signal and Array Processing, pp. 463–466 (2000)
5. Chen, V.C., Ling, H.: Time Frequency Transforms for Radar Imaging and Signal Analysis. Artech House, Boston (2002)
6. Anderson, M., Rogers, R.: Micro-Doppler Analysis of Multiple Frequency Continuous Wave Radar Signatures. In: SPIE Proc. Radar Sensor Technology, vol. 654 (2007)
7. Thayaparan, T., Abrol, S., Riseborough, E., Stankovic, L., Lamothe, D., Duff, G.: Analysis of Radar Micro-Doppler Signatures From Experimental Helicopter and Human Data. IEE Proc. Radar Sonar Navigation 1(4), 288–299 (2007)
8. Reynolds, D.A.A.: Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification. Ph.D.dissertation, Georgia Institute of Technology, Atlanta (1992)
9. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker Verification Using Adapted Gaussian Mixture Models. Digit. Signal Process. 10, 19–41 (2000)
10. Campbell, J.P.: Speaker Recognition: a tutorial. Proc.of the IEEE 85(9), 1437–1462 (1997)
11. Li, J.Q., Barron, A.R.: Mixture Density Estimation. In: Advances in Neural Information Processing Systems, p. 12. MIT Press, Cambridge (2002)
12. Bilik, I., Tabrikian, J., Cohen, A.: GMM-Based Target Classification for Ground Surveillance Doppler Radar. IEEE Trans. on Aerospace and Electronic Systems 42(1), 267–278 (2006)
13. Vander, H.F., Duin, W.R.P., de Ridder, D., Tax, D.M.J.: Classification, Parameter Estimation and State Estimation. John Wiley & Son, Ltd. (2004)

Speaker Identification Using Discriminative Learning of Large Margin GMM

Khalid Daoudi[1], Reda Jourani[2,3],
Régine André-Obrecht[2], and Driss Aboutajdine[3]

[1] GeoStat Group, INRIA Bordeaux-Sud Ouest, Talence, France
khalid.daoudi@inria.fr
[2] SAMoVA Group, IRIT - Univ. Paul Sabatier, Toulouse, France
{jourani,obrecht}@irit.fr
[3] Laboratoire LRIT. Faculty of Sciences, Mohammed 5 Agdal Univ., Rabat, Morocco
aboutaj@fsr.ac.ma

Abstract. Gaussian mixture models (GMM) have been widely and successfully used in speaker recognition during the last decades. They are generally trained using the generative criterion of maximum likelihood estimation. In an earlier work, we proposed an algorithm for discriminative training of GMM with diagonal covariances under a large margin criterion. In this paper, we present a new version of this algorithm which has the major advantage of being computationally highly efficient, thus well suited to handle large scale databases. We evaluate our fast algorithm in a Symmetrical Factor Analysis compensation scheme. We carry out a full NIST speaker identification task using NIST-SRE'2006 data. The results show that our system outperforms the traditional discriminative approach of SVM-GMM supervectors. A 3.5% speaker identification rate improvement is achieved.

Keywords: Large margin training, Gaussian mixture models, Discriminative learning, Speaker recognition, Session variability modeling.

1 Introduction

Most of state-of-the-art speaker recognition systems rely on the generative training of Gaussian Mixture Models (GMM) using maximum likelihood estimation and maximum a posteriori estimation (MAP) [1]. This generative training estimates the feature distribution within each speaker. In contrast, the discriminative training approaches model the boundary between speakers [2,3], thus generally leading to better performances than generative methods. For instance, Support Vector Machines (SVM) combined with GMM supervectors are among state-of-the-art approaches in speaker verification [4,5].

In speaker recognition applications, mismatch between the training and testing conditions can decrease considerably the performances. The inter-session variability, that is the variability among recordings of a given speaker, remains the most challenging problem to solve. The Factor Analysis techniques [6,7], e.g., Symmetrical Factor Analysis (SFA) [8], were proposed to address that problem

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 300–307, 2011.
© Springer-Verlag Berlin Heidelberg 2011

in GMM based systems. While the Nuisance Attribute Projection (NAP) [9] compensation technique is designed for SVM based systems.

Recently a new discriminative approach for multiway classification has been proposed, the Large Margin Gaussian mixture models (LM-GMM) [10]. The latter have the same advantage as SVM in term of the convexity of the optimization problem to solve. However they differ from SVM because they draw nonlinear class boundaries directly in the input space. While LM-GMM have been used in speech recognition, they have not been used in speaker recognition (to the best of our knowledge). In an earlier work [11], we proposed a simplified version of LM-GMM which exploit the fact that traditional GMM based speaker recognition systems use diagonal covariances and only the mean vectors are MAP adapted. We then applied this simplified version to a "small" speaker identification task. While the resulting training algorithm is more efficient than the original one, we found however that it is still not efficient enough to process large databases such as in NIST Speaker Recognition Evaluation (NIST-SRE) campaigns (*http://www.itl.nist.gov/iad/mig//tests/sre/*).

In order to address this problem, we propose in this paper a new approach for fast training of Large-Margin GMM which allow efficient processing in large scale applications. To do so, we exploit the fact that in general not all the components of the GMM are involved in the decision process, but only the k-best scoring components. We also exploit the property of correspondence between the MAP adapted GMM mixtures and the Universal Background Model mixtures [1]. In order to show the effectiveness of the new algorithm, we carry out a full NIST speaker identification task using NIST-SRE'2006 (core condition) data. We evaluate our fast algorithm in a Symmetrical Factor Analysis (SFA) compensation scheme, and we compare it with the NAP compensated GMM supervector Linear Kernel system (GSL-NAP) [5]. The results show that our Large Margin compensated GMM outperform the state-of-the-art discriminative approach GSL-NAP.

The paper is organized as follows. After an overview on Large-Margin GMM training with diagonal covariances in section 2, we describe our new fast training algorithm in section 3. The GSL-NAP system and SFA are then described in sections 4 and 5, respectively. Experimental results are reported in section 6.

2 Overview on Large Margin GMM with Diagonal Covariances (LM-dGMM)

In this section we start by recalling the original Large Margin GMM training algorithm developed in [10]. We then recall the simplified version of this algorithm that we introduced in [11].

In Large Margin GMM [10], each class c is modeled by a mixture of ellipsoids in the D-dimensional input space. The m^{th} ellipsoid of the class c is parameterized by a centroid vector μ_{cm}, a positive semidefinite (orientation) matrix Ψ_{cm} and a nonnegative scalar offset $\theta_{cm} \geq 0$. These parameters are then collected into a single enlarged matrix Φ_{cm}:

$$\Phi_{cm} = \begin{pmatrix} \Psi_{cm} & -\Psi_{cm}\mu_{cm} \\ -\mu_{cm}^T\Psi_{cm} & \mu_{cm}^T\Psi_{cm}\mu_{cm} + \theta_{cm} \end{pmatrix}. \quad (1)$$

A GMM is first fit to each class using maximum likelihood estimation. Let $\{o_{nt}\}_{t=1}^{T_n}$ ($o_{nt} \in \mathcal{R}^D$) be the T_n feature vectors of the n^{th} segment (i.e. n^{th} speaker training data). Then, for each o_{nt} belonging to the class y_n, $y_n \in \{1, 2, ..., C\}$ where C is the total number of classes, we determine the index m_{nt} of the Gaussian component of the GMM modeling the class y_n which has the highest posterior probability. This index is called *proxy label*. The training algorithm aims to find matrices Φ_{cm} such that "all" examples are correctly classified by at least one margin unit, leading to the LM-GMM criterion:

$$\forall c \neq y_n,\ \forall m, \qquad z_{nt}^T \Phi_{cm} z_{nt} \ \geq 1 + z_{nt}^T \Phi_{y_n m_{nt}} z_{nt}, \tag{2}$$

where $z_{nt} = [o_{nt} \ \ 1]^T$.

In speaker recognition, most of state-of-the art systems use diagonal covariances GMM. In these GMM based speaker recognition systems, a speaker-independent *world model* or *Universal Background Model* (UBM) is first trained with the EM algorithm. When enrolling a new speaker to the system, the parameters of the UBM are adapted to the feature distribution of the new speaker. It is possible to adapt all the parameters, or only some of them from the background model. Traditionally, in the GMM-UBM approach, the target speaker GMM is derived from the UBM model by updating only the mean parameters using a *maximum a posteriori* (MAP) algorithm [1].

Making use of this assumption of diagonal covariances, we proposed in [11] a simplified algorithm to learn GMM with a large margin criterion. This algorithm has the advantage of being more efficient than the original LM-GMM one [10] while it still yielded similar or better performances on a speaker identification task. In our Large Margin diagonal GMM (LM-dGMM) [11], each class (speaker) c is initially modeled by a GMM with M diagonal mixtures (trained by MAP adaptation of the UBM in the setting of speaker recognition). For each class c, the m^{th} Gaussian is parameterized by a mean vector μ_{cm}, a diagonal covariance matrix $\Sigma_m = diag(\sigma_{m1}^2, ..., \sigma_{mD}^2)$, and the scalar factor θ_m which corresponds to the weight of the Gaussian. For each example o_{nt}, the goal of the training algorithm is now to force the log-likelihood of its proxy label Gaussian m_{nt} to be at least one unit greater than the log-likelihood of each Gaussian component of all competing classes. That is, given the training examples $\{(o_{nt}, y_n, m_{nt})\}_{n=1}^N$, we seek mean vectors μ_{cm} which satisfy the LM-dGMM criterion:

$$\forall c \neq y_n,\ \forall m, \qquad d(o_{nt}, \mu_{cm}) + \theta_m \geq 1 + d(o_{nt}, \mu_{y_n m_{nt}}) + \theta_{m_{nt}}, \tag{3}$$

where $d(o_{nt}, \mu_{cm}) = \sum_{i=1}^{D} \frac{(o_{nti} - \mu_{cmi})^2}{2\sigma_{mi}^2}$. Afterward, these M constraints are fold into a single one using the softmax inequality $min_m\ a_m \geq -log \sum_m e^{-a_m}$. The segment-based LM-dGMM criterion becomes thus:

$$\forall c \neq y_n, \quad \frac{1}{T_n} \sum_{t=1}^{T_n} -log \sum_{m=1}^{M} e^{(-d(o_{nt}, \mu_{cm}) - \theta_m)} \geq 1 + \frac{1}{T_n} \sum_{t=1}^{T_n} d(o_{nt}, \mu_{y_n m_{nt}}) + \theta_{m_{nt}}. \tag{4}$$

Letting $[f]_+ = max(0, f)$ denote the so-called *hinge* function, the loss function to minimize for LM-dGMM is then given by:

$$\text{Ł} = \sum_{n=1}^{N} \sum_{c \neq y_n} \left[1 + \frac{1}{T_n} \sum_{t=1}^{T_n} \left(d(o_{nt}, \mu_{y_n m_{nt}}) + \theta_{m_{nt}} + log \sum_{m=1}^{M} e^{(-d(o_{nt}, \mu_{cm}) - \theta_m)} \right) \right]_+ . \tag{5}$$

3 LM-*d*GMM Training with *k*-Best Gaussians

3.1 Description of the New LM-dGMM Training Algorithm

Despite the fact that our LM-dGMM is computationally much faster than the original LM-GMM of [10], we still encountered efficiency problems when dealing with high number of Gaussian mixtures. In order to develop a fast training algorithm which could be used in large scale applications such as NIST-SRE, we propose to drastically reduce the number of constraints to satisfy in (4). By doing so, we would drastically reduce the computational complexity of the loss function and its gradient. To achieve this goal we propose to use another property of state-of-the-art GMM systems, that is, decision is not made upon all mixture components but only using the k-best scoring Gaussians. In other words, for each o_n and each class c, instead of summing over the M mixtures in the left side of (4), we would sum only over the k Gaussians with the highest posterior probabilities selected using the GMM of class c. In order to further improve efficiency and reduce memory requirement, we exploit the property reported in [1] about correspondence between MAP adapted GMM mixtures and UBM mixtures. We use the UBM to select one unique set S_{nt} of k-best Gaussian components per frame o_{nt}, instead of $(C-1)$ sets. This leads to a $(C-1)$ times faster and less memory consuming selection. More precisely, we now seek mean vectors μ_{cm} that satisfy the large margin constraints in (6):

$$\forall c \neq y_n, \quad \frac{1}{T_n} \sum_{t=1}^{T_n} -log \sum_{m \in S_{nt}} e^{(-d(o_{nt}, \mu_{cm}) - \theta_m)} \geq 1 + \frac{1}{T_n} \sum_{t=1}^{T_n} d(o_{nt}, \mu_{y_n m_{nt}}) + \theta_{m_{nt}}. \tag{6}$$

The resulting loss function expression is straightforward. During test, we use again the same principle to achieve fast scoring. Given a test segment of T frames, for each test frame x_t we use the UBM to select the set E_t of k-best scoring proxy labels and compute the LM-dGMM likelihoods using only these k labels. The decision rule is thus given as:

$$y = argmin_c \left\{ \sum_{t=1}^{T} -log \sum_{m \in E_t} e^{(-d(o_t, \mu_{cm}) - \theta_m)} \right\}. \tag{7}$$

3.2 Handling of Outliers

We adopt the strategy of [10] to detect outliers and reduce their negative effect on learning, by using the initial GMM models. We compute the accumulated hinge loss incurred by violations of the large margin constraints in (6):

$$h_n = \sum_{c \neq y_n} \left[1 + \frac{1}{T_n} \sum_{t=1}^{T_n} \left(d(o_{nt}, \mu_{y_n m_{nt}}) + \theta_{m_{nt}} + log \sum_{m \in S_{nt}} e^{(-d(o_{nt}, \mu_{cm}) - \theta_m)} \right) \right]_+ . \tag{8}$$

h_n measures the decrease in the loss function when an initially misclassified segment is corrected during the course of learning. We associate outliers with large values of h_n. We then re-weight the hinge loss terms by using the segment weights $s_n = min(1, 1/h_n)$:

$$\text{Ł} = \sum_{n=1}^{N} s_n h_n. \tag{9}$$

We solve this unconstrained non-linear optimization problem using the second order optimizer LBFGS [12].

4 The GSL-NAP System

In this section we briefly describe the GMM supervector linear kernel SVM system (GSL) [4] and its associated channel compensation technique, the Nuisance attribute projection (NAP) [9].

Given an M-components GMM adapted by MAP from the UBM, one forms a GMM supervector by stacking the D-dimensional mean vectors. This GMM supervector (an MD vector) can be seen as a mapping of variable-length utterances into a fixed-length high-dimensional vector, through GMM modeling:

$$\phi(\mathrm{x}) = [\mu_{x1} \ \cdots \ \mu_{xM}]^T, \tag{10}$$

where the GMM $\{\mu_{xm}, \Sigma_m, w_m\}$ is trained on the utterance x. For two utterances x and y, a kernel distance based on the Kullback-Leibler divergence between the GMM models trained on these utterances [4], is defined as:

$$\mathrm{K}(\mathrm{x}, \mathrm{y}) = \sum_{m=1}^{M} \left(\sqrt{w_m} \Sigma_m^{-(1/2)} \mu_{xm} \right)^T \left(\sqrt{w_m} \Sigma_m^{-(1/2)} \mu_{ym} \right). \tag{11}$$

The UBM weight and variance parameters are used to normalize the Gaussian means before feeding them into a linear kernel SVM training. This system is referred to as GSL in the rest of the paper.

NAP is a pre-processing method that aims to compensate the supervectors by removing the directions of undesired sessions variability, before the SVM training [9]. NAP transforms a supervector ϕ to a compensated supervector $\hat{\phi}$:

$$\hat{\phi} = \phi - \mathbf{S}(\mathbf{S}^T \phi), \tag{12}$$

using the eigenchannel matrix $\mathbf{S}$, which is trained using several recordings (sessions) of various speakers. Given a set of expanded recordings of N different speakers, with h_i different sessions for each speaker s_i, one first removes the speakers variability by subtracting the mean of the supervectors within each speaker. The resulting supervectors are then pooled into a single matrix $\mathbf{C}$ representing the intersession variations. One identifies finally the subspace of dimension R where the variations are the largest by solving the eigenvalue problem on the covariance matrix $\mathbf{CC}^T$, getting thus the projection matrix $\mathbf{S}$ of a size $MD \times R$. This system is referred to as GSL-NAP in the rest of the paper.

5 Symmetrical Factor Analysis (SFA)

In this section we describe the symmetrical variant of the Factor Analysis model (SFA) [8] (Factor Analysis was originally proposed in [6,7]).

In the mean supervector space, a speaker model can be decomposed into three different components: a session-speaker independent component (the UBM model), a speaker dependent component and a session dependent component. The session-speaker model, can be written as [8]:

$$\mathbf{M}_{(h,s)} = \mathbf{M} + \mathbf{D}\mathbf{y}_s + \mathbf{U}\mathbf{x}_{(h,s)}, \tag{13}$$

where

- $\mathbf{M}_{(h,s)}$ is the session-speaker dependent supervector mean (an MD vector),
- $\mathbf{M}$ is the UBM supervector mean (an MD vector),
- $\mathbf{D}$ is a $MD \times MD$ diagonal matrix, where $\mathbf{DD}^T$ represents the a priori covariance matrix of $\mathbf{y}_s$,
- $\mathbf{y}_s$ is the speaker vector, i.e., the speaker offset (an MD vector),
- $\mathbf{U}$ is the session variability matrix of low rank R (an $MD \times R$ matrix),
- $\mathbf{x}_{(h,s)}$ are the channel factors, i.e., the session offset (an R vector not dependent on s in theory).

$\mathbf{D}\mathbf{y}_s$ and $\mathbf{U}\mathbf{x}_{(h,s)}$ represent respectively the speaker dependent component and the session dependent component.

The factor analysis modeling starts by estimating the $\mathbf{U}$ matrix, using different recordings per speaker. Given the fixed parameters $(\mathbf{M}, \mathbf{D}, \mathbf{U})$, the target models are then compensated by eliminating the session mismatch directly in the model domain. Whereas, the compensation in the test is performed at the frame level (feature domain).

6 Experimental Results

We perform experiments on the NIST-SRE'2006 speaker identification task and compare the performances of the baseline GMM, the LM-dGMM and the SVM systems, with and without using channel compensation techniques. The comparisons are made on the male part of the NIST-SRE'2006 core condition (1conv4w-1conv4w). The feature extraction is carried out by the filter-bank based cepstral

Table 1. Speaker identification rates with GMM, Large Margin diagonal GMM and GSL models, with and without channel compensation

System	256 Gaussians	512 Gaussians
GMM	76.46%	77.49%
LM-dGMM	77.62%	78.40%
GSL	81.18%	82.21%
LM-dGMM-SFA	89.65%	**91.27%**
GSL-NAP	87.19%	87.77%

analysis tool Spro [13]. Bandwidth is limited to the 300-3400Hz range. 24 filter bank coefficients are first computed over 20ms Hamming windowed frames at a 10ms frame rate and transformed into Linear Frequency Cepstral Coefficients (LFCC). Consequently, the feature vector is composed of 50 coefficients including 19 LFCC, their first derivatives, their 11 first second derivatives and the delta-energy. The LFCCs are preprocessed by Cepstral Mean Subtraction and variance normalization. We applied an energy-based voice activity detection to remove silence frames, hence keeping only the most informative frames. Finally, the remaining parameter vectors are normalized to fit a zero mean and unit variance distribution.

We use the state-of-the-art open source software ALIZE/Spkdet [14] for GMM, SFA, GSL and GSL-NAP modeling. A male-dependent UBM is trained using all the telephone data from the NIST-SRE'2004. Then we train a MAP adapted GMM for the 349 target speakers belonging to the primary task. The corresponding list of 539554 trials (involving 1546 test segments) are used for test. Score normalization techniques are not used in our experiments. The so MAP adapted GMM define the baseline GMM system, and are used as initialization for the LM-dGMM one. The GSL system uses a list of 200 impostor speakers from the NIST-SRE'2004, on the SVM training. The LM-dGMM-SFA system is initialized by model domain compensated GMM, which are then discriminated using feature domain compensated data. The session variability matrix $\mathbf{U}$ of SFA and the channel matrix $\mathbf{S}$ of NAP, both of rank $R = 40$, are estimated on NIST-SRE'2004 data using 2934 utterances of 124 different male speakers.

Table 1 shows the speaker identification accuracy scores of the various systems, for models with 256 and 512 Gaussian components ($M = 256, 512$). All these scores are obtained with the 10 best proxy labels selected using the UBM, $k = 10$. The results of Table 1 show that, without SFA channel compensation, the LM-dGMM system outperforms the classical generative GMM one, however it does yield worse performances than the discriminative approach GSL. Nonetheless, when applying channel compensation techniques, GSL-NAP outperforms GSL as expected, but the LM-dGMM-SFA system significantly outperforms the GSL-NAP one. Our best system achieves 91.27% speaker identification rate, while the best GSL-NAP achieves 87.77%. This leads to a 3.5% improvement. These results show that our fast Large Margin GMM discriminative learning algorithm not only allows efficient training but also achieves better speaker identification accuracy than a state-of-the-art discriminative technique.

7 Conclusion

We presented a new fast algorithm for discriminative training of Large-Margin diagonal GMM by using the k-best scoring Gaussians selected form the UBM. This algorithm is highly efficient which makes it well suited to process large scale databases. We carried out experiments on a full speaker identification task under the NIST-SRE'2006 core condition. Combined with the SFA channel compensation technique, the resulting algorithm significantly outperforms the state-of-the-art speaker recognition discriminative approach GSL-NAP. Another major advantage of our method is that it outputs diagonal GMM models. Thus, broadly used GMM techniques/softwares such as SFA or ALIZE/Spkdet can be readily applied in our framework. Our future work will consist in improving margin selection and outliers handling. This should indeed improve the performances.

References

1. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker Verification Using Adapted Gaussian Mixture Models. Digit. Signal Processing 10(1-3), 19–41 (2000)
2. Keshet, J., Bengio, S.: Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods. Wiley, Hoboken (2009)
3. Louradour, J., Daoudi, K., Bach, F.: Feature Space Mahalanobis Sequence Kernels: Application to Svm Speaker Verification. IEEE Trans. Audio Speech Lang. Processing 15(8), 2465–2475 (2007)
4. Campbell, W.M., Sturim, D.E., Reynolds, D.A.: Support Vector Machines Using GMM Supervectors for Speaker Verification. IEEE Signal Processing Lett. 13(5), 308–311 (2006)
5. Campbell, W.M., Sturim, D.E., Reynolds, D.A., Solomonoff, A.: SVM Based Speaker Verification Using a GMM Supervector Kernel and NAP Variability Compensation. In: ICASSP, vol. 1, pp. I-97–I-100 (2006)
6. Kenny, P., Boulianne, G., Dumouchel, P.: Eigenvoice Modeling with Sparse Training Data. IEEE Trans. Speech Audio Processing 13(3), 345–354 (2005)
7. Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P.: Speaker and Session Variability in GMM-Based Speaker Verification. IEEE Trans. Audio Speech Lang. Processing 15(4), 1448–1460 (2007)
8. Matrouf, D., Scheffer, N., Fauve, B.G.B., Bonastre, J.-F.: A Straightforward and Efficient Implementation of the Factor Analysis Model for Speaker Verification. In: Interspeech, pp. 1242–1245 (2007)
9. Solomonoff, A., Campbell, W.M., Boardman, I.: Advances in Channel Compensation for SVM Speaker Recognition. In: ICASSP, vol. 1, pp. 629–632 (2005)
10. Sha, F., Saul, L.K.: Large Margin Gaussian Mixture Modeling for Phonetic Classification and Recognition. In: ICASSP, vol. 1, pp. 265–268 (2006)
11. Jourani, R., Daoudi, K., André-Obrecht, R., Aboutajdine, D.: Large Margin Gaussian Mixture Models for Speaker Identification. In: Interspeech, pp. 1441–1444 (2010)
12. Nocedal, J., Wright, S.: Numerical Optimization. Springer, New York (1999)
13. Gravier, G.: SPro: Speech Signal Processing Toolkit (2003), https://gforge.inria.fr/projects/spro
14. Bonastre, J.-F., et al.: ALIZE/SpkDet: a State-of-the-art Open Source Software for Speaker Recognition. In: Odyssey, paper 020 (2008)

Sparse Coding Image Denoising Based on Saliency Map Weight

Haohua Zhao and Liqing Zhang*

MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, China
zhang-lq@cs.sjtu.edu.cn

Abstract. Saliency maps provide a measurement of people's attention to images. People pay more attention to salient regions and perceive more information in them. Image denoising enhances image quality by reducing the noise in contaminated images. Here we implement an algorithm framework to use a saliency map as weight to manage tradeoffs in denoising using sparse coding. Computer simulations confirm that the proposed method achieves better performance than a method without the saliency map.

Keywords: sparse coding, saliency map, image denoise.

1 Introduction

Saliency maps provide a measurement of people's attention to images. People pay more attention to salient regions and perceive more information in them. Many algorithms have been developed to generate saliency maps. [7] first introduced the maps, and [4] improved the method. Our team has also implemented some saliency map algorithms such as [5], [6].

Sparse coding provides a new approach to image denoising. Several important algorithms have been implemented. [2] and [1] provide an algorithm using KSVD to learn the sparse basis (dictionary) and reconstruct the image. In [9], a constraint that the similar patches have to have a similar sparse coding has been added to the sparse model for denoising. [8] introduce a method that uses an overcomplete topographical method to learn a dictionary and denoise the image. In these methods, if some of the parameters were changed, we would get more detail from the denoised images, but with more noise. In some regions in an image, people want to preserve more detail and do not care so much about the remaining noise but not in other regions.

Salient regions in an image usually contain more abundant information than nonsalient regions. Therefore it is reasonable to weight those regions heavily in order to achieve better accuracy in the reconstructed image. In image denoising,

* Corresponding Author.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 308–315, 2011.
© Springer-Verlag Berlin Heidelberg 2011

the more detail preserved, the more noise remains. We use the salience as weight to optimize this tradeoff.

In this paper, we will use *sparse coding with saliency map* and *image reconstruction with saliency map* to make use of saliency maps in image denoising. Computer simulations will be used to show the performance of the proposed method.

2 Saliency Map

There are many approaches to defining the saliency map of an image. In [6], results depend on the given sparse basis so that is not suitable for denoising. In [5], if a texture appears in many places in an image, then these places do not get large salience values. The result of [4] is too central for our algorithm. This impairs the performance of our algorithm. The result of [7] is suitable enough to implement in our approach since it is not affected by the noise and the large salience distributes are not so central as [4]. Therefore we use this method to get the saliency map $S(x)$, normalized to the interval $[0, 1]$. Here we used the code published on [3], which can produce the saliency maps in [7] and [4].

We add Gaussian white noise with variance $\sigma = 25$ on an image in our database (results in Fig.1(a)) and compute the saliency map which is in Fig.1(b). We can see that we got a very good saliency result for the denoising tradeoff problem.

The histogram of the saliency map in Fig.1(b) is shown in Fig.1(c). Many of the saliency values are in the range $[0, 0.3]$, which is not suitable for our next operation, so we apply a transform to the saliency values. Calling the median saliency m_e, the transform is:

$$S_m(x) = [S(x) + (1 - \beta m_e)]^\theta ; \tag{2.1}$$

Where $\beta > 0$ and $\theta \in \mathbb{R}$ are constants. After the transform, we get:

$$S_m(x) \begin{cases} = 1 & \text{if } S(x) = \beta m_e \\ > 1 & \text{if } S(x) < \beta m_e \\ < 1 \text{ and } \geq 0 & \text{if } S(x) > \beta m_e \end{cases} \tag{2.2}$$

Set $S_m(x_1) > 1$, $0 \leq S_m(x_{-1}) < 1$, and $S_m(x_0) = 1$. $S_m(x_1)$ gets larger, $S_m(x_{-1})$ gets smaller, and $S_m(x_0)$ does not change if θ gets larger. Otherwise it gets the inverse. This helps us a lot in our following operation.

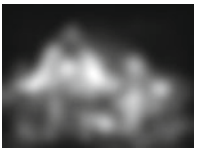
To make our next operation simpler, we use the function in [3] to resize the map to the same as the input image, and processes a Gaussian filter on it if the noise is preserved in the map[1], as (2.3) shows, where G_3 is the function to do this.

$$\tilde{S}(x) = G_3 [S_m(x)] \tag{2.3}$$

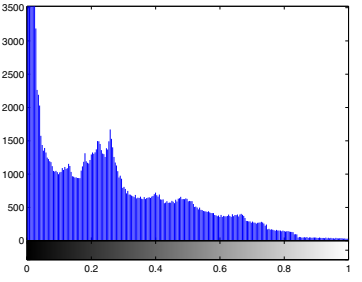
[1] We didn't use this filter in our experiment since the maps do not contain noise.



(a) Noisy image



(b) Saliency map



(c) Histogram

Fig. 1. A noisy image, its saliency map and the histogram of the saliency map

3 Sparse Coding with Saliency

First, we get some 8×8 patches from the image. In our method, we assume that the sparse basis is already known. The dictionary can be learned by the algorithm in [1] or [3]. In our approach, we use the DCT (Discrete Cosine Transform) basis as dictionary for simplicity. The following uses the sparse coefficients of this basis to represent the patches (we call it *sparse coding*). We use the OMP sparse algorithm in [10] because it is fast and effective.

In the OMP algorithm, we want to solve the optimization problem

$$\begin{aligned} &\min \|\alpha\|_0 \\ &\text{s.t.} \|\mathbf{Y} - \mathbf{D}\alpha\| < \delta, (\delta > 0) \end{aligned} \tag{3.1}$$

Where $\mathbf{Y}$ is the original image batch, $\mathbf{D}$ is the dictionary, α is the coding coefficient. In [2], $\delta = C\sigma$, where C is a constant that is set to 1.15, and σ is the noise variance.

When δ gets smaller, we get more detail after sparse coding. So we can use the saliency value as a parameter to change δ.

$$\delta'(X) = \frac{\delta}{\tilde{S}(X) + \varepsilon} \tag{3.2}$$

Where $\varepsilon > 0$ is a small constant that makes the denominator not be 0. X is the image patch to deal with. Let x be a pixel in X. We define $\tilde{S}(X) = \text{mean}\left\{\underset{x \in X}{\tilde{S}}(x)\right\}$.

Then the optimization problem is changed to (3.3)

$$\begin{aligned} &\min \|\alpha\|_0 \\ &\text{s.t.} \|\mathbf{Y} - \mathbf{D}\alpha\| < \delta'(X) = \frac{\delta}{\tilde{S}(X) + \varepsilon} \end{aligned} \tag{3.3}$$

Set $\tilde{S}(X_1)+\varepsilon>1$, $\tilde{S}(X_{-1})+\varepsilon<1$, and $\tilde{S}(X_0)+\varepsilon=1$. We can conclude that the areas can be sorted as $X_1>X_0>X_{-1}$ by the attention people pay to them. From (3.3), we will get $\delta'(X_1)<\delta'(X_0)<\delta'(X_{-1})$, which tells us the detail we get from X_1 is more than X_0, which is the same as the original method and more than X_{-1}. At the same time, the patch X_{-1} will become smoother and have less noise as we want.

4 Image Reconstruction with Saliency

After getting the sparse coding, we can do the image reconstruction. We do this based on the denoising algorithms in [2] but without learning the Dictionary (the sparse basis) for adapting the noisy image using K-SVD[1].

In [2], the image reconstruction process is to solve the optimization problem.

$$\hat{\mathbf{X}}=\underset{\mathbf{X}}{\operatorname{argmin}}\left\{\lambda\|\mathbf{X}-\mathbf{Y}\|_2^2+\sum_{ij}\|\mathbf{D}\hat{\boldsymbol{\alpha}}_{ij}-\mathbf{R}_{ij}\mathbf{X}\|_2^2\right\} \tag{4.1}$$

Where $\mathbf{Y}$ is the noisy image, $\mathbf{D}$ is the sparse dictionary, $\hat{\boldsymbol{\alpha}}_{ij}$ is the patch ij's sparse coefficients, which we know or have computed, $\mathbf{R}_{ij}$ are the matrices that turn the image into patches. λ is a constant to trade off the two items. In [2], $\lambda=30/\sigma$.

In (4.1), the first item minimizes the difference between the noisy image and the denoised image; the second item minimizes the difference between the image using the sparse coding and the denoised image. We can conclude that the first item minimizes the loss of detail while the second minimizes the noise. We can make use of the salience here; we change the optimization problem into (4.2)

$$\hat{\mathbf{X}}=\underset{\mathbf{X}}{\operatorname{argmin}}\left\{\lambda\|\mathbf{X}-\mathbf{Y}\|_2^2+\sum_{ij}\tilde{S}(\mathbf{Y}_{ij})^{-\gamma}\|\mathbf{D}\hat{\boldsymbol{\alpha}}_{ij}-\mathbf{R}_{ij}\mathbf{X}\|_2^2\right\} \tag{4.2}$$

Where $\gamma\geq 0$. Then the solution will be as (4.3)

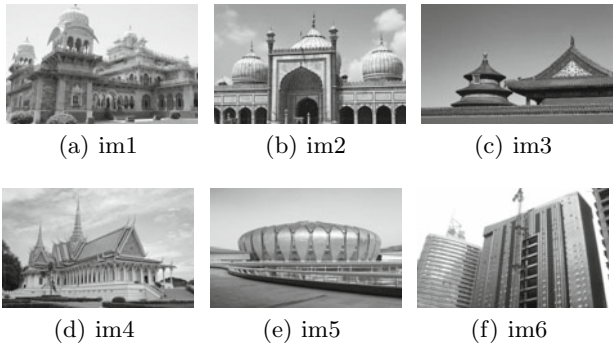
$$\hat{\mathbf{X}}=\left(\lambda\mathbf{I}+\sum_{ij}\tilde{S}(\mathbf{Y}_{ij})^{-\gamma}\mathbf{R}_{ij}^T\mathbf{R}_{ij}\right)^{-1}\left(\lambda\mathbf{Y}+\sum_{ij}\tilde{S}(\mathbf{Y}_{ij})^{-\gamma}\mathbf{R}_{ij}^T\mathbf{D}\hat{\boldsymbol{\alpha}}_{ij}\right) \tag{4.3}$$

5 Experiment and Result

5.1 Experiment

Here we tried using only a sparse coding with saliency (equivalent to setting $\gamma=0$), using only image reconstruction with saliency (equivalent to setting $\theta=0$ and $\varepsilon=0$), and using both methods (equivalent to setting $\gamma>0$, $\theta>0$) to check the performance of our algorithm . We will show the denoised result

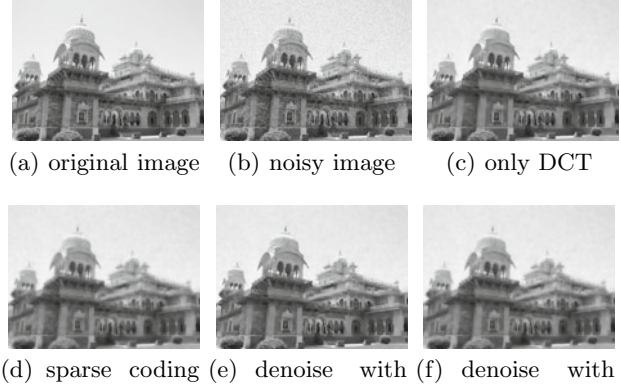
of the image shown in Fig.1(a) (See Fig.3). Then we will list the PSNR (Peak signal-to-noise ratio) of the result of the images in Fig.2, which are downloaded from the Internet and all have a building with some texture and a smooth sky. Also we will show the result of DCT denoising in [2] with DCT basis as basis for comparison. We will try to analyze the advantages and the disadvantages of our method based on the experimental results. Some detail of the global parameters is as follows: $C = 1.15$; $\lambda = 30/\sigma$, $\beta = 0.5$, $\theta = 1$, $\gamma = 4$.



(a) im1 (b) im2 (c) im3

(d) im4 (e) im5 (f) im6

Fig. 2. Test Images



(a) original image (b) noisy image (c) only DCT

(d) sparse coding with saliency (e) denoise with saliency (f) denoise with both methods

Fig. 3. Denoise result of the image in Fig.1(a)

Only sparse coding with saliency. A result image is shown in Fig.3(d). Here we try some other images and change σ of the noise. We can see how the result changes in Table 1. Unfortunately PSNR is smaller than the original DCT denoising, especially when σ is small. However, when σ gets larger, the PSNRs get closer to the original DCT method (See Fig.4).

Table 1. Result (PSNR (dB)) of the images in Fig.2

	σ	5	15	25	50	75
im1	sparse coding with salience	29.5096	27.9769	26.7156	24.7077	23.4433
	image reconstruction with saliency	38.1373	31.2929	28.5205	25.2646	23.6842
	Both method	30.6479	28.2903	26.8357	24.6799	23.3490
	only DCT	38.1896	31.2696	28.4737	25.2263	23.6629
im2	sparse coding with salience	26.5681	25.4787	24.4215	22.3606	20.9875
	image reconstruction with saliency	37.5274	30.6464	27.6311	23.6068	21.4183
	Both method	27.9648	25.9360	24.6235	22.3926	20.9744
	only DCT	37.5803	30.6546	27.6070	23.5581	21.3736
im3	sparse coding with salience	29.5156	28.4537	27.3627	25.2847	23.9346
	image reconstruction with saliency	39.9554	32.7652	29.6773	25.9388	24.1149
	Both method	30.9932	28.9424	27.5767	25.3047	23.9068
	only DCT	40.0581	32.7738	29.6525	25.8998	24.0833
im4	sparse coding with salience	28.8955	27.4026	26.1991	24.2200	22.9965
	image reconstruction with saliency	37.8433	31.3429	28.5906	25.0128	23.2178
	Both method	29.9095	27.7025	26.3360	24.2459	22.9836
	only DCT	37.8787	31.3331	28.5600	24.9753	23.1880
im5	sparse coding with salience	30.6788	29.1139	27.7872	25.4779	23.9669
	image reconstruction with saliency	39.5307	33.0688	30.2126	26.2361	24.0337
	Both method	31.7282	29.4005	27.8970	25.4685	23.9195
	only DCT	39.6354	33.0814	30.2007	26.2157	24.0131
im6	sparse coding with salience	26.8868	25.4964	24.3416	22.3554	21.1347
	image reconstruction with saliency	37.5512	30.6229	27.5820	23.4645	21.4496
	Both method	27.9379	25.8018	24.4768	22.3709	21.1165
	only DCT	37.6788	30.6474	27.5773	23.4368	21.4252
Aver.	sparse coding with salience	28.6757	27.3204	26.1380	24.0677	22.7439
	image reconstruction with saliency	38.4242	31.6232	28.7024	24.9206	22.9864
	Both method	29.8636	27.6789	26.2910	24.0771	22.7083
	only DCT	38.5035	31.6267	28.6785	24.8853	22.9577

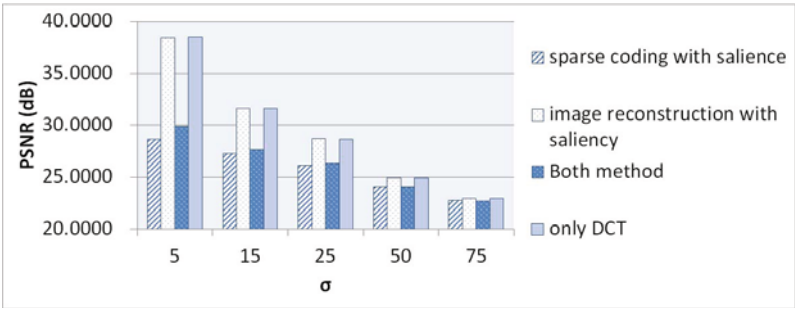


Fig. 4. Average denoise result

But in running the program, we found that the time cost for our method is less than the original method when most of $\tilde{S}(X)$ are smaller than 1... This is because the sparse stage uses most of the time, and as δ gets larger, time gets smaller. In our method, most of $\tilde{S}(X)$ are smaller than 1 if we set $\beta \geq 1$, which would not change the result much, we can save time in the sparse stage. Computing the saliency map does not cost much time.
Generally speaking, our purpose has been realized here. We preserved more detail in the regions that have larger salience values.

Only reconstructing image with saliency. A result image can see Fig.3(e). We can see that the result has been improved. More results are in Table 1 and Fig.4. When $\sigma \geq 25$, the PSNRs is better than the original method... But when $\sigma < 25$, the PSNRs become smaller.

Both methods. The result image is in Fig.3(f). The PSNRs of the denoised result for images in Fig.2 are in Table 1 and Fig..4. We can see that in this case, the result has combined the features of the two methods. The PSNRs are better than only using sparse coding with saliency, but not as good as the original method and image reconstruction with saliency. However, the time cost is also small.

5.2 Result Discussion

As we mentioned above, in some cases our method will cost less time than the original DCT denoising. Also, using image reconstruction with saliency in the images with heavy noise, our method perform better than the original DCT denoising.

From Fig.3, we can see that in our approach the sky, which has low saliency and little detail, has been blurred, which is what we want, and some detail of the building is preserved, though some noise and some strange texture caused by the basis is left there. We can change the parameters, such as θ, C, γ, and λ, to make the background smoother or preserve more detail (however, more noise) for the foreground.

We do better in blurring the background than preserving the foreground detail now. Sometimes when preserving the foreground detail, too much noise remains in the result image, and the gray value of the regions with different saliency seems not well-matched. In other words, the edge between this region is too strong. But for this problem we have already used the function G_3 to get an artial solution.

6 Discussion

In this paper, we introduce a method using a saliency map in image denoising with sparse coding. We use this to improve the tradeoff between the detail and the noise in the image.

The attention people pay to images generally fits the salience value, but some people may focus on different regions of the image in some cases. We can try different saliency map approaches in our framework to meet this requirement.

How to pick the patches may be very important in the denoising approach. In the current approach, we just pick all the patches or pick a patch every several pixels. In the future, we can try to pick more patches in the region where the salience value is large.

Since there is some strange texture in the denoised image because of the basis, we can try to use a learned dictionary, as in the algorithm in [8], which seems to be more suitable for natural scenes.

Acknowledgement. The work was supported by the National Natural Science Foundation of China (Grant No. 90920014) and the NSFC-JSPS International Cooperation Program (Grant No. 61111140019) .

Reference

1. Aharon, M., Elad, M., Bruckstein, A.: k-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing 54(11), 4311–4322 (2006)
2. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Processing 15(12), 3736–3745 (2006)
3. Harel, J.: Saliency map algorithm: Matlab source code, http://www.klab.caltech.edu/~harel/share/gbvs.php
4. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. Advances in Neural Information Processing Systems 19, 545 (2007)
5. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–8 (June 2007)
6. Hou, X., Zhang, L.: Dynamic visual attention: Searching for coding length increments. Advances in Neural Information Processing Systems 21, 681–688 (2008)
7. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(11), 1254–1259 (1998)
8. Ma, L., Zhang, L.: A hierarchical generative model for overcomplete topographic representations in natural images. In: International Joint Conference on Neural Networks, IJCNN 2007, pp. 1198–1203 (August 2007)
9. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: 2009 IEEE 12th International Conference on Computer Vision, September 29-October 2, pp. 2272–2279 (2009)
10. Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S.: Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: 1993 Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 40–44 (November 1993)

Expanding Knowledge Source with Ontology Alignment for Augmented Cognition

Jeong-Woo Son, Seongtaek Kim, Seong-Bae Park,
Yunseok Noh, and Jun-Ho Go

School of Computer Science and Engineering,
Kyungpook National University, Korea
{jwson,stkim,sbpark,ysnoh,jhgo}@sejong.knu.ac.kr

Abstract. Augmented cognition on sensory data requires knowledge sources to expand the abilities of human senses. Ontologies are one of the most suitable knowledge sources, since they are designed to represent human knowledge and a number of ontologies on diverse domains can cover various objects in human life. To adopt ontologies as knowledge sources for augmented cognition, various ontologies for a single domain should be merged to prevent noisy and redundant information. This paper proposes a novel composite kernel to merge heterogeneous ontologies. The proposed kernel consists of lexical and graph kernels specialized to reflect structural and lexical information of ontology entities. In experiments, the composite kernel handles both structural and lexical information on ontologies more efficiently than other kernels designed to deal with general graph structures. The experimental results also show that the proposed kernel achieves the comparable performance with top-five systems in OAEI 2010.

1 Introduction

Augmented cognition aims to amplify human capabilities such as strength, decision making, and so on [11]. Among various human capabilities, the senses are one of the most important things, since they provide basic information for other capabilities. Augmented cognition on sensory data aims to expand information from human senses. Thus, it requires additional knowledges. Among Various knowledge sources, ontologies are the most appropriate knowledge source, since they represent human knowledges on a specific domain in a machine-readable form [9] and a mount of ontologies which cover diverse domains are publicly available.

One of the issues related with ontologies as knowledge sources is that most ontologies are written separately and independently by human experts to serve particular domains. Thus, there could be many ontologies even in a single domain, and it causes semantic heterogeneity. The heterogeneous ontologies for a domain can provide redundant or noisy information. Therefore, it is demanded to merge related ontologies to adopt ontologies as a knowledge source for augmented cognition on sensory data.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 316–324, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Ontology alignment aims to merge two or more ontologies which contain similar semantic information by identifying semantic similarities between entities in the ontologies. An ontology entity has two kinds of information: lexical information and structural information. Lexical information is expressed in labels or values of some properties.The lexical similarity is then easily designed as a comparison of character sequences in labels or property values. The structure of an entity is, however, represented as a graph due to its various relations with other entities. Therefore, a method to compare graphs is needed to capture the structural similarity between entities.

This paper proposes a composite kernel function for ontology alignment. The composite kernel function is composed of a lexical kernel based on Levenshtein distance for lexical similarity and a graph kernel for structural similarity. The graph kernel in the proposed composite kernel is a modified version of the random walk graph kernel proposed by Gätner et al. [6]. When two graphs are given, the graph kernel implicitly enumerates all possible entity random walks, and then the similarity between the graphs is computed using the shared entity random walks.

Evaluation of the composite kernel is done with the Conference data set from OAEI (Ontology Alignment Evaluation Initiative) 2010 campaign[1]. It is shown that the ontology kernel is superior to the random walk graph kernel in matching performance and computational cost. In comparison with OAEI 2010 competitors, it achieves a comparable performance.

2 Related Work

Various structural similarities have been designed for ontology alignment [3]. ASMOV, one of the state-of-the-art alignment system, computes a structural similarity by decomposing an entity graph into two subgraphs [8]. These two subgraphs contain relational and internal structure respectively. From the relational structure, a similarity is obtained by comparing ancestor-descendant relations, while relations from object properties are reflected by the internal structures. OMEN [10] and iMatch [1] use a network-based model. They first approximate roughly the probability that two ontology entities match using lexical information, and then refine the probability by performing probabilistic reasoning over the entity network. The main drawback of most previous work is that structural information is expressed in some specific forms such as a label-path, a vector, and so on rather than a graph itself. This is because a graph is one of the most difficult data structures to compare. Thus, whole structural information of all nodes and edges in the graph is not reflected in computing structural similarity.

Haussler [7] proposed a solution to this problem, so-called convolution kernel which determines the similarity between structural data such as tree, graph, and so on by shared sub-structures. Since the structure of an ontology entity can be regarded as a graph, the similarity between entities can be obtained by a convolution kernel for a graph. The random walk graph kernel proposed by

[1] http://oaei.ontologymatching.org/2010

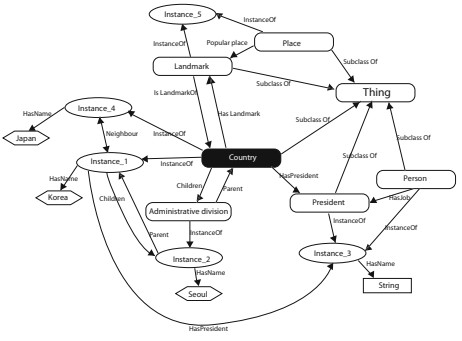


Fig. 1. An example of ontology graph

Gärtner et al. [6] is commonly used for ordinary graph structures. In this kernel, random-walks are regarded as sub-structures. Thus, the similarity of two graphs is computed by measuring how many random-walks are shared. Graph kernels can compare graphs without any structural transformation [2].

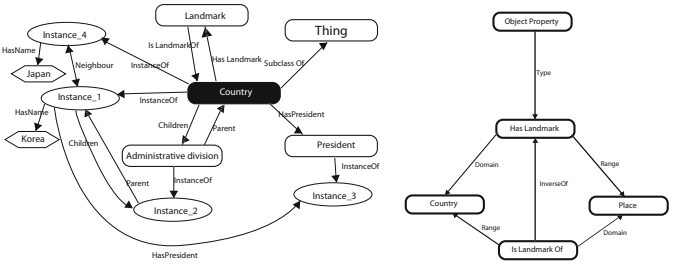
2.1 Ontology as Graph

An ontology is regarded as a graph of which nodes and edges are ontology entities [12]. Figure 1 shows a simple ontology with a domain of topography. As shown in this figure, nodes are generated from four ontology entities: concepts, instances, property value types, and property values. Edges are generated from object type properties and data type properties.

3 Ontology Alignment

A concept of an ontology has a structure, since it has relations with other entities. Thus, it can be regarded as a subgraph of the *ontology graph*. The subgraph for a concept is called as *concept graph*. Figure 2(a) shows the concept graph for a concept, *Country* on the ontology in Figure 1. A property also has a structure, the *property graph* to describe the structure of a property. Unlike the concept graph, in the property graph, the target property becomes a node. All concepts and properties also become nodes if they restrict the property with an axiom. The axioms used to restrict them are edges of the graph. Figure 2(b) shows the property graph for a property, *Has Location*.

One of the important characteristic in both concept and property graphs is that all nodes and edges have not only their labels but also their types like concept, instance and so on. Since some concepts can be defined properties and, at the same time, some properties can be represented as concepts in ontologies, these types are importance to characterize the structure of concept and property graphs,



(a) Concept graph (b) Property graph

Fig. 2. An example of concept and property graphs

3.1 Ontology Alignment with Similarity

Let E_i be a set of concepts and properties in an ontology O_i. The alignment of two ontologies O_1 and O_2 aims to generate a list of concept-to-concept and property-to-property pairs [5]. In this paper, it is assumed that many entities from O_2 can be matched to an entity in O_1. Then, all entities in E_2 whose similarity with $e_1 \in E_1$ is larger than a pre-defined threshold θ become the matched entities of e_1. That is, for an entity $e_1 \in E_1$, a set E_2^* is matched which satisfies

$$E_2^* = \{e_2 \in E_2 | sim(e_1, e_2) \geq \theta\}. \tag{1}$$

Note that the key factor of Equation (1) is obviously the similarity, $sim(e_1, e_2)$.

4 Similarity between Ontology Entities

The entity of an ontology is represented with two types of information: lexical and structural information. Thus, an entity e_i can be represented as $e_i =< L_{e_i}, G_{e_i} >$ where L_{e_i} denotes the label of e_i, while G_{e_i} is the graph structure for e_i. The similarity function, of course, should compare both lexical and structural information.

4.1 Graph Kernel

The main obstacle of computing $sim(G_{e_i}, G_{e_j})$ is the graph structure of entities. Comparing two graphs is a well-known problem in the machine learning community. One possible solution to this problem is a *graph kernel*. A graph kernel maps graphs into a feature space spanned by their subgraphs. Thus, for given two graphs G_1 and G_2, the kernel is defined as

$$K_{graph}(G_1, G_2) = \Phi(G_1) \cdot \Phi(G_2), \tag{2}$$

where Φ is a mapping function which maps a graph onto a feature space.

A random walk graph kernel uses all possible random walks as features of graphs. Thus, all random walks should be enumerated in advance to compute the similarity. Gätner et al. [6] adopted a direct product graph as a way to avoid explicit enumeration of all random walks. The direct product graph of G_1 and G_2 is denoted by $G_1 \times G_2 = (V_\times, E_\times)$, where $V_\times$ and $E_\times$ are the node and edge sets that are defined respectively as

$$\begin{aligned} V_\times(G_1 \times G_2) &= \{(v_1, v_2) \in V_1 \times V_2 : l(v_1) = l(v_2)\}, \\ E_\times(G_1 \times G_2) &= \{((v_1, v_1'), (v_2, v_2')) \in V_\times(G_1 \times G_2) : \\ &\quad (v_1, v_1') \in E_1 \text{ and } (v_2, v_2') \in E_2 \\ &\quad \text{and } l(v_1, v_1') = l(v_2, v_2')\}, \end{aligned}$$

where $l(v)$ is the label of a node v and $l(v, v')$ is the label of an edge between two nodes v and v'.

From the adjacency matrix $A \in \mathbb{R}^{|V_\times| \times |V_\times|}$ of $G_1 \times G_2$, the similarity of G_1 and G_2 can be directly computed without explicit enumeration of all random walks. The adjacency matrix A has a well-known characteristic. When the adjacency matrix is multiplied n times, an element $A^n_{v_\times, v'_\times}$ becomes the summation of similarities between random walks of length n from $v_\times$ to $v'_\times$, where $v_\times \in V_\times$ and $v'_\times \in V_\times$. Thus, by adopting a direct product graph and its adjacency matrix, Equation (2) is rewritten as

$$K_{graph}(G_1, G_2) = \sum_{i,j=1}^{|V_\times|} \left[\sum_{n=0}^{\infty} \lambda_n A^n \right]_{i,j}. \tag{3}$$

4.2 Modified Graph Kernel

Even though the graph kernel efficiently determines a similarity between graphs with their shared random walks, it can not reflect the characteristics of graphs for ontology entities. In both concept and property graphs, nodes and edges represents not only their labels but also their types. To reflect this characteristic, a modified version of the graph kernel is proposed in this paper. In the modified graph kernel, the direct product graph is defined as $G_1 \times G_2 = (V^o_\times, E^o_\times)$, where $V^o_\times$ and $E^o_\times$ are re-defined as

$$\begin{aligned} V^o_\times(G_1 \times G_2) &= \{(v_1, v_2) \in V_1 \times V_2 : l(v_1) = l(v_2) \text{ and } t(v_1) = t(v_2)\}, \\ E^o_\times(G_1 \times G_2) &= \{((v_1, v_1'), (v_2, v_2')) \in V^o_\times(G_1 \times G_2) : \\ &\quad (v_1, v_1') \in E_1 \text{ and } (v_2, v_2') \in E_2, \\ &\quad l(v_1, v_1') = l(v_2, v_2') \text{ and } t(v_1, v_1') = t(v_2, v_2')\}, \end{aligned}$$

where $t(v)$ and $t(v, v')$ are types of the node v and the edge (v, v') respectively.

The modified graph kernel can simply adopt types of nodes and edges in a similarity. The adjacency matrix A in the modified graph kernel has smaller size than that in the random walk graph kernel. Since nodes in concept and

property graphs are composed of concept, property, instance and so on, the size of $V_\times$ in the graph kernel is $|V_\times| = \sum_{t \in T} n_t(G_1) \cdot \sum_{t \in T} n_t(G_2)$, where T is a set of types appeared in ontologies and $n_t(G)$ returns the number of nodes with type t in the graph G. However, the modified graph kernel uses $V_\times^o$ with the size of $|V_\times^o| = \sum_{t \in T} n_t(G_1) \cdot n_t(G_2)$. The computational cost of the graph kernel is $O(l \cdot |V_\times|^3)$ where l is the maximum length of random walks. Accordingly, by adopting types of nodes and edges, the modified graph kernel prunes away nodes with different types from the direct product graph. It results in less computational cost than one of the random walk graph kernel.

4.3 Composite Kernel

An entity of an ontology is represented with structural and lexical information. Graphs for structural information of entities are compared with the modified graph kernel, while similarities between labels for lexical information of entities is determined a lexical kernel. In this paper, a lexical kernel is designed by using inverse of Levenshtein distance between entity labels. A similarity between a pair of entities with both information is obtained by using a composite kernel, $K_C(e_i, e_j) = \frac{K_G(G_{e_i}, G_{e_j}) + K_L(L_{e_i}, L_{e_j})}{2}$, where $K_G()$ denotes the modified graph kernel and $K_L()$ is the lexical kernel. In the composite kernel both information are reflected with the same importance.

5 Experiments

5.1 Experimental Data and Setting

Experiments are performed with Conference data set constructed by Ontology Alignment Evaluation Initiative (OAEI). This data set has seven real world ontologies describing organizing conferences and 21 reference alignments among them are given. The ontologies have only concepts and properties and the average number of concepts is 72, and that of properties is 44.42.

In experiments, all parameters are set heuristically. The maximum length of random walks in both the random walk and modified graph kernels is two, and θ in Equation (1) is 0.70 for the modified graph kernel and 0.79 for the random walk graph kernel.

5.2 Experimental Result

Table 1 shows the performances of three different kernels: the modified graph kernel, the random walk graph kernel, and the lexical kernel. LD denotes Levenshtein distance, while GK and MGK are the random walk graph kernel and the modified graph kernel respectively. As shown in this table, GK shows the worst performance, F-measure of 0.41 and it implies that graphs of ontology entities have different characteristics from ordinary graphs. MGK can reflects the characteristics on graphs of ontology entities. Consequently, MGK achieves the best

Table 1. The performance of the modified graph kernel, the lexical kernel and the random walk graph kernel

Method	Precision	Recall	F-measure
LK	0.62	0.41	0.49
GK	0.47	0.37	0.41
MGK	**0.84**	**0.42**	**0.56**

Table 2. The performances of composite kernels

Method	Precision	Recall	F-measure
LK+GK	0.49	0.45	0.46
LK+MGK	**0.74**	**0.49**	**0.59**

performance, F-measure of 0.56 and it is 27% improvement in F-measure over GK. LK does not shows good performance due to lack of structural information. Even though LK does not shows good performance, it reflects the different aspect of entities from both graph kernels. Therefore, there exists a room to improve by combining LK with a graph kernel.

Table 2 shows the performances of composite kernels to reflect both structural and lexical information. In this table, the proposed composite kernel (LK+MGK) is compared with a composite kernel (LK+GK) composed of the lexical kernel and the random walk graph kernel. As shown in this table, for all evaluation measures, LK+MGK shows better performances than LK+GK. Even though LK+MGK shows less precision than one of MGK, it achieves better recall and F-measure. The experimental results implies that structural and lexical information of entities should be considered in entity comparison and the proposed composite kernel efficiently handles both information.

Figure 3 shows computation times of both modified and random walk graph kernels. In this experiment, the computation times are measured on a PC running Microsoft Windows Server 2008 with Intel Core i7 3.0 GHz processor and 8 GB RAM. In this figure, X-axis refers to ontologies in Conference data set and Y-axis is average computation time. Since each ontology is matched six times with the other ontologies, the time in Y-axis is the average of the six matching times. For all ontologies, the modified kernel demands just a quarter computation time of the random walk graph kernel. The random walk graph kernel uses about 3,150 seconds on average, but the modified graph kernel spends just 830 seconds on average by pruning the adjacent matrix. The results of the experiments prove that the modified graph kernel is more efficient for ontology alignment than the random walk graph kernel from the viewpoints of both performance and computation time.

Table 3 compares the proposed composite kernel with OAEI 2010 competitors [4]. As shown in this table, the proposed kernel shows the performance within top-five performances. The best system in OAEI 2010 campaign is CODI which depends on logics generated by human experts. Since it relies on the hand-crafted logics, it suffers from low recall. ASMOV and Eff2Match adopts various

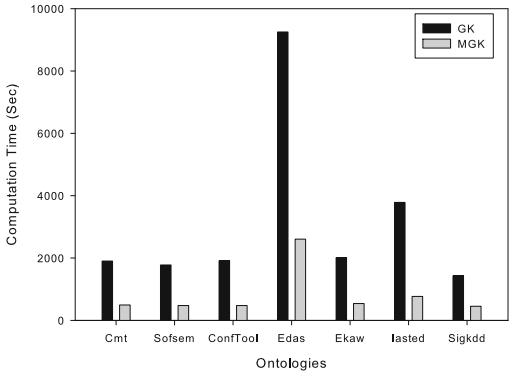


Fig. 3. The computation times of the ontology kernel and the random walk graph kernel

Table 3. The performances of OAEI 2010 participants and the ontology kernel

	AgrMaker	AROMA	ASMOV	CODI	Eff2Match
Precision	0.53	0.36	0.57	0.86	0.61
Recall	0.62	0.49	0.63	0.48	0.60
F-measure	0.58	0.42	0.60	0.62	0.60
	Falcon	GeRMeSMB	COBOM	LK+MGK	
Precision	0.74	0.37	0.56	**0.74**	
Recall	0.49	0.51	0.56	**0.49**	
F-measure	0.59	0.43	0.56	**0.59**	

similarities for generality. Thus, the precisions of both systems are below the precision of the proposed kernel.

6 Conclusion

Augmented cognition on sensory data demands knowledge sources to expand sensory information. Among various knowledge sources, ontologies are the most appropriate one, since they are designed to represent human knowledge in a machine-readable form and there exist a number of ontologies on diverse domains. To adopt ontologies as a knowledge source for augmented cognition, various ontologies on the same domain should be merged to reduce redundant and noisy information.

For this purpose, this paper proposed a novel composite kernel to compare ontology entities. The proposed composite kernel is composed of the modified graph kernel and the lexical kernel. From the fact that all entities such as concepts and properties in the ontology are represented as a graph, the modified version of the random walk graph kernel is adopted to efficiently compares structures of ontology entities. The lexical kernel determines a similarity between entities with their

lexical information. As a result, the composite kernel can reflect both structural and lexical information of ontology entities.

In a series of experiments, we verified that the modified graph kernel handles structural information of ontology entities more efficiently than the random walk graph kernel from the viewpoints of performance and computation time. It also shows that the proposed composite kernel can efficiently handle both structural and lexical information. In comparison with the competitors of OAEI 2010 campaign, the composite kernel achieved the comparable performance with OAEI 2010 competitors.

Acknowledgement. This research was supported by the Converging Research Center Program funded by the Ministry of Education, Science and Technology (2011K000659).

References

1. Albagli, S., Ben-Eliyahu-Zohary, R., Shimony, S.: Markov network based ontology matching. In: Proceedings of the 21th IJCAI, pp. 1884–1889 (2009)
2. Costa, F., Grave, K.: Fast neighborhood subgraph pairwise distance kernel. In: Proceedings of the 27th ICML, pp. 255–262 (2010)
3. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)
4. Euzenat, J., Ferrara, A., Meilicke, C., Pane, J., Scharffe, F., Shvaiko, P., Stuckenschmidt, H., Šváb Zamazal, O., Svátek, V., Santos, C.: First results of the ontology alignment evaluation initiative 2010. In: Proceedings of OM 2010, pp. 85–117 (2010)
5. Euzenat, J.: Semantic precision and recall for ontology alignment evaluation. In: Proceedings of the 20th IJCAI, pp. 348–353 (2007)
6. Gärtner, T., Flach, P., Wrobel, S.: On Graph Kernels: Hardness Results and Efficient Alternatives. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 129–143. Springer, Heidelberg (2003)
7. Haussler, D.: Convolution kernels on discrete structures. Technical report, UCS-CRL-99-10, UC Santa Cruz (1999)
8. Jean-Mary, T., Shironoshita, E., Kabuka, M.: Ontology matching with semantic verification. Journal of Web Semantics 7(3), 235–251 (2009)
9. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intelligent Systems 16(2), 72–79 (2001)
10. Mitra, P., Noy, N., Jaiswal, A.R.: OMEN: A Probabilistic Ontology Mapping Tool. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 537–547. Springer, Heidelberg (2005)
11. Schmorrow, D.: Foundations of Augmented Cognition. Human Factors and Ergonomics (2005)
12. Shvaiko, P., Euzenat, J.: A Survey of Schema-Based Matching Approaches. In: Spaccapietra, S. (ed.) Journal on Data Semantics IV. LNCS, vol. 3730, pp. 146–171. Springer, Heidelberg (2005)

Nyström Approximations for Scalable Face Recognition: A Comparative Study

Jeong-Min Yun[1] and Seungjin Choi[1,2]

[1] Department of Computer Science
[2] Division of IT Convergence Engineering
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu, Pohang 790-784, Korea
{azida,seungjin}@postech.ac.kr

Abstract. Kernel principal component analysis (KPCA) is a widely-used statistical method for representation learning, where PCA is performed in reproducing kernel Hilbert space (RKHS) to extract nonlinear features from a set of training examples. Despite the success in various applications including face recognition, KPCA does not scale up well with the sample size, since, as in other kernel methods, it involves the eigen-decomposition of $n \times n$ Gram matrix which is solved in $\mathcal{O}(n^3)$ time. Nyström method is an approximation technique, where only a subset of size $m \ll n$ is exploited to approximate the eigenvectors of $n \times n$ Gram matrix. In this paper we consider Nyström method and its few modifications such as 'Nyström KPCA ensemble' and 'Nyström + randomized SVD' to improve the scalability of KPCA. We compare the performance of these methods in the task of learning face descriptors for face recognition.

Keywords: Face recognition, Kernel principal component analysis, Nyström approximation, Randomized singular value decomposition.

1 Introduction

Face recognition is a challenging pattern classification problem, the goal of which is to learn a classifier which automatically identifies unseen face images (see [9] and references therein). One of key ingredients in face recognition is how to extract fruitful face image descriptors. Subspace analysis is the most popular techniques, demonstrating its success in numerous visual recognition tasks such as face recognition, face detection and tracking. Singular value decomposition (SVD) and principal component analysis (PCA) are representative subspace analysis methods which were successfully applied to face recognition [7].

Kernel PCA (KPCA) is an extension of PCA, allowing for nonlinear feature extraction, where the linear PCA is carried out in reproducing kernel Hilbert space (RKHS) with a nonlinear feature mapping [6]. Despite the success in various applications including face recognition, KPCA does not scale up well with the sample size, since, as in other kernel methods, it involves the eigen-decomposition

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 325–334, 2011.
© Springer-Verlag Berlin Heidelberg 2011

of $n \times n$ Gram matrix, $\boldsymbol{K}_{n,n} \in \mathbb{R}^{n \times n}$, which is solved in $\mathcal{O}(n^3)$ time. Nyström method approximately computes the eigenvectors of the Gram matrix $\boldsymbol{K}_{n,n}$ by carrying out the eigendecomposition of an $m \times m$ block, $\boldsymbol{K}_{m,m} \in \mathbb{R}^{m \times m}$ $(m \ll n)$ and expanding these eigenvectors back to n dimensions using the information on the thin block $\boldsymbol{K}_{n,m} \in \mathbb{R}^{n \times m}$.

In this paper we consider the Nyström approximation for KPCA and its modifications such as 'Nyström KPCA ensemble' that is adopted from our previous work on landmark MDS ensemble [3] and 'Nyström + randomized SVD' [4] to improve the scalability of KPCA. We compare the performance of these methods in the task of learning face descriptors for face recognition.

2 Methods

2.1 KPCA in a Nutshell

Suppose that we are given n samples in the training set, so that the data matrix is denoted by $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$, where $\boldsymbol{x}_i$'s are the vectorized face images of size d. We consider a feature space $\mathcal{F}$ induced by a nonlinear mapping $\phi(\boldsymbol{x}_i) : \mathbb{R}^d \mapsto \mathcal{F}$. Transformed data matrix is given by $\boldsymbol{\Phi} = [\phi(\boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x}_n)] \in \mathbb{R}^{r \times n}$. The Gram matrix (or kernel matrix) is given by $\boldsymbol{K}_{n,n} = \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \in \mathbb{R}^{n \times n}$. Define the centering matrix by $\boldsymbol{H} = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ where $\mathbf{1}_n \in \mathbb{R}^n$ is the vector of ones and $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$ is the identity matrix. Then the centered Gram matrix is given by $\widetilde{\boldsymbol{K}}_{n,n} = (\boldsymbol{\Phi}\boldsymbol{H})^\top(\boldsymbol{\Phi}\boldsymbol{H})$. On the other hand, the data covariance matrix in the feature space is given by $\boldsymbol{C}_\phi = (\boldsymbol{\Phi}\boldsymbol{H})(\boldsymbol{\Phi}\boldsymbol{H})^\top = \boldsymbol{\Phi}\boldsymbol{H}\boldsymbol{\Phi}^\top$ since $\boldsymbol{H}$ is symmetric and idempotent, i.e., $\boldsymbol{H}^2 = \boldsymbol{H}$. KPCA seeks k leading eigenvectors $\boldsymbol{W} \in \mathbb{R}^{r \times k}$ of $\boldsymbol{C}_\phi$ to compute the projections $\boldsymbol{W}^\top(\boldsymbol{\Phi}\boldsymbol{H})$.

To this end, we consider the following eigendecomposition:

$$(\boldsymbol{\Phi}\boldsymbol{H})(\boldsymbol{\Phi}\boldsymbol{H})^\top \boldsymbol{W} = \boldsymbol{W}\boldsymbol{\Sigma}. \tag{1}$$

Pre-multiply both sides of (1) by $(\boldsymbol{\Phi}\boldsymbol{H})^\top$ to obtain

$$(\boldsymbol{\Phi}\boldsymbol{H})^\top(\boldsymbol{\Phi}\boldsymbol{H})(\boldsymbol{\Phi}\boldsymbol{H})^\top \boldsymbol{W} = (\boldsymbol{\Phi}\boldsymbol{H})^\top \boldsymbol{W}\boldsymbol{\Sigma}. \tag{2}$$

From the representer theorem, we assume $\boldsymbol{W} = \boldsymbol{\Phi}\boldsymbol{H}\boldsymbol{U}$, and then plug in this relation into (2) to obtain

$$(\boldsymbol{\Phi}\boldsymbol{H})^\top(\boldsymbol{\Phi}\boldsymbol{H})(\boldsymbol{\Phi}\boldsymbol{H})^\top \boldsymbol{\Phi}\boldsymbol{H}\boldsymbol{U} = (\boldsymbol{\Phi}\boldsymbol{H})^\top \boldsymbol{\Phi}\boldsymbol{H}\boldsymbol{U}\boldsymbol{\Sigma}, \tag{3}$$

leading to

$$\widetilde{\boldsymbol{K}}_{n,n}^2 \boldsymbol{U} = \widetilde{\boldsymbol{K}}_{n,n}\boldsymbol{U}\boldsymbol{\Sigma}, \tag{4}$$

the solution to which is determined by solving the simplified eigenvalue equation:

$$\widetilde{\boldsymbol{K}}_{n,n}\boldsymbol{U} = \boldsymbol{U}\boldsymbol{\Sigma}. \tag{5}$$

Note that column vectors in $\boldsymbol{U}$ in (5) are normalized such that $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{\Sigma}^{-1}$ to satisfy $\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_k$, then normalized eigenvectors are denoted by $\widetilde{\boldsymbol{U}} = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{U}$. Given l test data points, $\boldsymbol{X}_* \in \mathbb{R}^{d\times l}$, the projections onto the eigenvectors $\boldsymbol{W}$ are computed by

$$\begin{aligned}\boldsymbol{Y}_* &= \boldsymbol{W}^\top \left(\boldsymbol{\Phi}_* - \boldsymbol{\Phi}\frac{1}{n}\mathbf{1}_n\mathbf{1}_l^\top\right) = \widetilde{\boldsymbol{U}}^\top \left(\boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right)\boldsymbol{\Phi}^\top \left(\boldsymbol{\Phi}_* - \boldsymbol{\Phi}\frac{1}{n}\mathbf{1}_n\mathbf{1}_l^\top\right) \\ &= \widetilde{\boldsymbol{U}}^\top \left(\boldsymbol{K}_{n,l} - \boldsymbol{K}_{n,n}\frac{1}{n}\mathbf{1}_n\mathbf{1}_l^\top - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{K}_{n,l} + \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{K}_{n,n}\frac{1}{n}\mathbf{1}_n\mathbf{1}_l^\top\right), \quad (6)\end{aligned}$$

where $\boldsymbol{K}_{n,l} = \boldsymbol{\Phi}^\top\boldsymbol{\Phi}_*$.

2.2 Nyström Approximation for KPCA

A bottleneck in KPCA is in computing the eigenvectors of $\widetilde{\boldsymbol{K}}_{n,n}$, which is solved in $\mathcal{O}(n^3)$ time. We select $m(\ll n)$ landmark points, or sample points, from $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ and partition the data matrix into $\boldsymbol{X}_m \in \mathbb{R}^{d\times m}$ (landmark data matrix) and $\boldsymbol{X}_{n-m} \in \mathbb{R}^{d\times(n-m)}$ (non-landmark data matrix), so that $\boldsymbol{X} = [\boldsymbol{X}_m, \boldsymbol{X}_{n-m}]$. Similarly we have $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_m, \boldsymbol{\Phi}_{n-m}]$. Centering $\boldsymbol{\Phi}$ leads to $\widetilde{\boldsymbol{\Phi}} = \boldsymbol{\Phi}\boldsymbol{H} = [\widetilde{\boldsymbol{\Phi}}_m, \widetilde{\boldsymbol{\Phi}}_{n-m}]$. Thus we partition the Gram matrix $\widetilde{\boldsymbol{K}}_{n,n}$ as

$$\widetilde{\boldsymbol{K}}_{n,n} = \begin{bmatrix} \widetilde{\boldsymbol{\Phi}}_m^\top\widetilde{\boldsymbol{\Phi}}_m & \widetilde{\boldsymbol{\Phi}}_m^\top\widetilde{\boldsymbol{\Phi}}_{n-m} \\ \widetilde{\boldsymbol{\Phi}}_{n-m}^\top\widetilde{\boldsymbol{\Phi}}_m & \widetilde{\boldsymbol{\Phi}}_{n-m}^\top\widetilde{\boldsymbol{\Phi}}_{n-m} \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{K}}_{m,m} & \widetilde{\boldsymbol{K}}_{m,n-m} \\ \widetilde{\boldsymbol{K}}_{n-m,m} & \widetilde{\boldsymbol{K}}_{n-m,n-m} \end{bmatrix}. \quad (7)$$

Denote $\boldsymbol{U}^{(m)} \in \mathbb{R}^{m\times k}$ as k leading eigenvectors of the $m \times m$ block $\widetilde{\boldsymbol{K}}_{m,m}$, i.e., $\widetilde{\boldsymbol{K}}_{m,m}\boldsymbol{U}^{(m)} = \boldsymbol{U}^{(m)}\boldsymbol{\Sigma}^{(m)}$. Nyström approximation [8] permits the computation of eigenvectors $\boldsymbol{U}$ and eigenvalues $\boldsymbol{\Sigma}$ of $\widetilde{\boldsymbol{K}}_{n,n}$ using $\boldsymbol{U}^{(m)}$ and $\widetilde{\boldsymbol{K}}_{n,m}^\top = [\widetilde{\boldsymbol{K}}_{m,m}^\top, \widetilde{\boldsymbol{K}}_{n-m,m}^\top]$:

$$\boldsymbol{U} \approx \sqrt{\frac{m}{n}}\widetilde{\boldsymbol{K}}_{n,m}\widetilde{\boldsymbol{K}}_{m,m}^{-1}\boldsymbol{U}^{(m)}, \quad \boldsymbol{\Sigma} \approx \frac{n}{m}\boldsymbol{\Sigma}^{(m)}. \quad (8)$$

2.3 Nyström KPCA Ensemble

Nyström approximation uses a single subset of size m to approximately compute the eigenvectors of $n \times n$ Gram matrix. Here we describe 'Nyström KPCA ensemble' where we combine individual Nyström KPCA solutions which operate on different partitions of the input. Originally this ensemble method was developed for landmark multidimensional scaling [3]. We consider one primal subset of size m and L subsidiary subsets, each of which is of size $m_L \leq m$. Given the input $\boldsymbol{X} \in \mathbb{R}^{d\times n}$ and the centered kernel matrix $\widetilde{\boldsymbol{K}}_{n,n}$, we denote by $\widehat{\boldsymbol{Y}}_i$ for $i = 0, 1, \ldots, L$ kernel projections onto Nyström approximations to eigenvectors:

$$\widehat{\boldsymbol{Y}}_i = \boldsymbol{\Sigma}_i^{-1/2}\boldsymbol{U}_i^\top\widetilde{\boldsymbol{K}}_{n,n}, \quad (9)$$

where $\boldsymbol{U}_i$ and $\boldsymbol{\Sigma}_i$ for $i = 0, 1, \ldots, L$, are Nyström approximations to eigenvectors and eigenvalues of $\widetilde{\boldsymbol{K}}_{n,n}$ computed using the primal subset ($i = 0$) and L subsidiary subsets.

Each solution $\widehat{\boldsymbol{Y}}_i$ is in different coordinate system. Thus, these solutions are aligned in a common coordinate system by affine transformations using ground control points (GCPs) that are shared by the primal and subsidiary subsets. We denote $\widehat{\boldsymbol{Y}}_0^c$ by the kernel projections of GCPs in the primal subset and choose it as reference. To line up $\widehat{\boldsymbol{Y}}_i$'s in a common coordinate, we determine affine transformations which satisfy

$$\begin{bmatrix} \boldsymbol{A}_i & \boldsymbol{\alpha}_i \\ \boldsymbol{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{Y}}_i^c \\ \boldsymbol{1}_p^\top \end{bmatrix} = \begin{bmatrix} \widehat{\boldsymbol{Y}}_0^c \\ \boldsymbol{1}_p^\top \end{bmatrix}, \tag{10}$$

for $i = 1, \ldots, L$ and p is the number of GCPs. Then, aligned solutions are computed by

$$\widetilde{\boldsymbol{Y}}_i = \boldsymbol{A}_i \widehat{\boldsymbol{Y}}_i + \boldsymbol{\alpha}_i \boldsymbol{1}_p^\top, \tag{11}$$

for $i = 1, \ldots, L$. Note that $\widetilde{\boldsymbol{Y}}_0 = \widehat{\boldsymbol{Y}}_0$. Finally we combine these aligned solutions with weights proportional to the number of landmark points:

$$\boldsymbol{Y} = \frac{m}{m + Lm_L} \widehat{\boldsymbol{Y}^0} + \frac{m_L}{m + Lm_L} \sum_{i=1}^{L} \widetilde{\boldsymbol{Y}^i}. \tag{12}$$

Nyström KPCA ensemble considers multiple subsets which may cover most of data points in the training set. Therefore, we can alternatively compute KPCA solutions without Nyström approximations

$$\widehat{\boldsymbol{Y}}_i^{(m)} = [\boldsymbol{\Sigma}_i^{(m)}]^{-1/2} [\boldsymbol{U}_i^{(m)}]^\top \widetilde{\boldsymbol{K}}_{m,n}, \tag{13}$$

where $\boldsymbol{U}_i^{(m)}$ and $\boldsymbol{\Sigma}_i^{(m)}$ are eigenvectors and eigenvalues of $m \times m$ or $m_L \times m_L$ kernel matrices involving the primal subset ($i = 0$) and L subsidiary subsets. One may follow the alignment and combination steps described above to compute the final solution.

2.4 Nyström + Randomized SVD

Randomized singular value decomposition (rSVD) is another type of the approximation algorithm of SVD or eigen-decomposition which is designed for fixed-rank case [1]. Given rank k and the matrix $\boldsymbol{K} \in \mathbb{R}^{n \times n}$, rSVD works with k-dimensional subspace of $\boldsymbol{K}$ instead of $\boldsymbol{K}$ itself by projecting it onto $n \times k$ random matrix, and this randomness enable the subspace to span the range of $\boldsymbol{K}$. (Detailed algorithm is shown in Algorithm 1.) Since the time complexity of rSVD is $O(n^2k + k^3)$, it runs very fast with small k. However, rSVD cannot be applied to very large data set because of $O(n^2k)$ term, so in recent, the combined method of rSVD and Nyström has been proposed [4] which achieves the time complexity of $O(nmk + k^3)$. We call it "rSVD + Nyström" for further references. The time complexities for KPCA, Nyström method, and its variants mentioned above are shown in Table 1 [3,4].

Algorithm 1. Randomized SVD for a symmetric matrix [1]

Input: $n \times n$ symmetric matrix $\boldsymbol{K}$, scalars k, p, q.
Output: Eigenvectors $\boldsymbol{U}$, eigenvalues $\boldsymbol{\Sigma}$.
1: Generate an $n \times (k+p)$ Gaussian random matrix $\boldsymbol{\Omega}$.
2: $\boldsymbol{Z} = \boldsymbol{K}\boldsymbol{\Omega}$, $\widehat{\boldsymbol{Z}} = \boldsymbol{K}^{q-1}\boldsymbol{Z}$.
3: Compute an orthonormal matrix $\boldsymbol{Q}$ by applying QR decomposition to $\widehat{\boldsymbol{Z}}$.
4: Compute an SVD of $\boldsymbol{Q}^\top \boldsymbol{K}$: $(\boldsymbol{Q}^\top \boldsymbol{K}) = \widehat{\boldsymbol{U}}\boldsymbol{\Sigma}\boldsymbol{V}$.
5: $\boldsymbol{U} = \boldsymbol{Q}\widehat{\boldsymbol{U}}$.

Table 1. The time complexities for variant methods. For ensemble methods, the sample size of each solutions is assume to be equal.

Method	Time complexity	Parameter
KPCA	$O(n^3)$	n: # of data points
Nyström	$O(nmk + m^3)$	m: # of sample points
rSVD	$O(n^2k + k^3)$	k: # of principal components
rSVD + Nyström	$O(nmk + k^3)$	L: # of solutions
Nyström KPCA ensemble	$O(Lnmk + Lm^3 + Lkp^2)$	p: # of GCPs

3 Numerical Experiments

We use frontal face images in XM2VTS database [5]. The data set consists of one set with 1,180 color face images of 295 people $\times$ 4 images at resolution 720×576, and the other set with 1,180 images for same people but take shots on another day. We use one set for the training set, the other for the test set. Using the eyes, nose, and mouth position information available in XM2VTS database web-site, we make the cropped image of each image, which focuses on the face and has same eyes position with each others. Finally, we convert each mage to a 64×64 grayscale image, and then apply Gaussian kernel with $\sigma^2 = 5$.

We consider the simple classification method: comparing correlation coefficients. Let $\widehat{\boldsymbol{x}}_i$ and $\widehat{\boldsymbol{y}}_j$ denote the data points after feature extraction in the training set and test set, respectively. ρ_{ij} is referred to their correlation coefficient, and if $l(\boldsymbol{x})$ is defined as a function returning $\boldsymbol{x}$'s class label, then

$$l(\widehat{\boldsymbol{y}}_j) = l(\widehat{\boldsymbol{x}}_{i^*}), \text{ where } i^* = \arg\max_i \rho_{ij} \tag{14}$$

3.1 Random Sampling with Class Label Information

Because our goal is to construct the large scale face recognition system, we basically consider the random sampling techniques for sample selection of the Nyström method. [2] report that uniform sampling without replacement is better than the other complicated non-uniform sampling techniques. For the face recognition system, class label information of the training set is available, then how about use this information for sampling? We call this way "sampling with

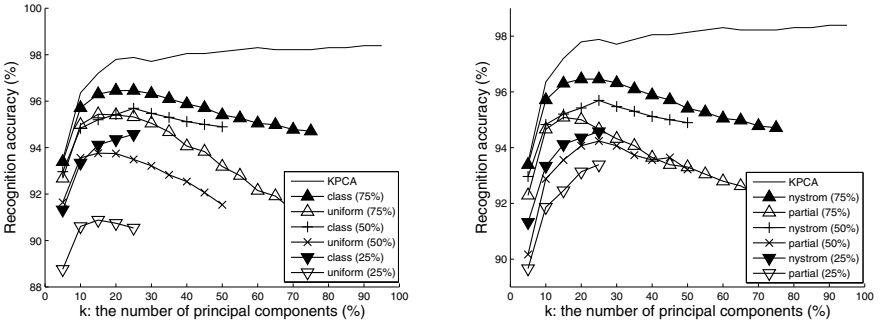


Fig. 1. Face recognition accuracy of KPCA and its Nyström approximation against variable m and k. (a) compares "uniform" sampling and sampling with "class" information. (b) compares full step "Nyström" method and "partial" one.

class information" and it can be done as follows. First, group all data points with respect to their class labels. Then randomly sample a point of each group in rotation until the desired number of samples are collected.

As you can see in Fig. 1 (a), sampling with class information always produces better face recognition accuracy than uniform sampling. The result makes sense if we assume that the data points in the same class tend to cluster together, and this assumption is the typical assumption of any kind of classification problems. For the following experiments, we use a "sampling with class information" technique.

3.2 Is Nyström Really Helpful for Face Recognition?

In Nyström approximation, we get two different sets of eigenvectors. First one is m-dimensional eigenvectors obtained from $\widetilde{\boldsymbol{K}}_{m,m}$. Another one is n-dimensional eigenvectors which are approximate eigenvectors of the original Gram matrix.

Since the standard Nyström method is designed to approximate the Gram matrix, m-dimensional eigenvectors have only been used as intermediate results. In face recognition, however, the objective is to extract features, so they also can be used as feature vectors. Then, do approximate n-dimensional eigenvectors give better results than m-dimensional ones? Fig. 1 (b) answers it.

We denote feature extraction with n-dimensional eigenvectors as a full step Nyström method, and extraction with m-dimensional ones as a partial step. And the figure shows that the full step gives about 1% better accuracy than the partial one among three different sample sizes. The result may come from the usage of additional part of the Gram matrix in the full step Nyström method.

3.3 How Many Samples/Principal Components are Needed?

In this section, we test the effect of the sample size m and the number of principal components k (Fig. 2 (a)). For m, we test seven different sample sizes, and

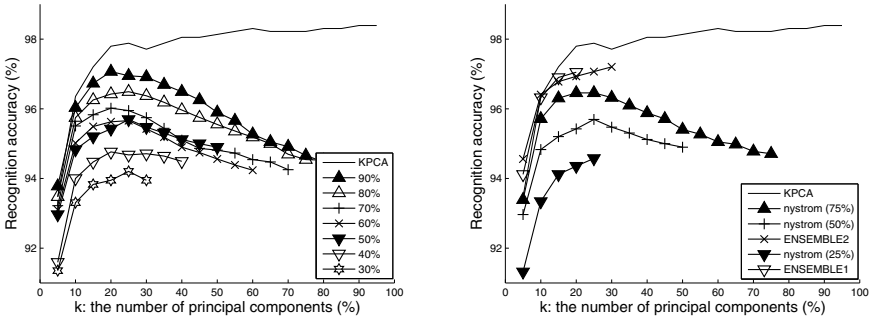


Fig. 2. (a) Face recognition accuracy of KPCA and its Nyström approximation against variable m and k. (b) Face recognition accuracy of KPCA, its Nyström approximation, and Nyström KPCA ensemble.

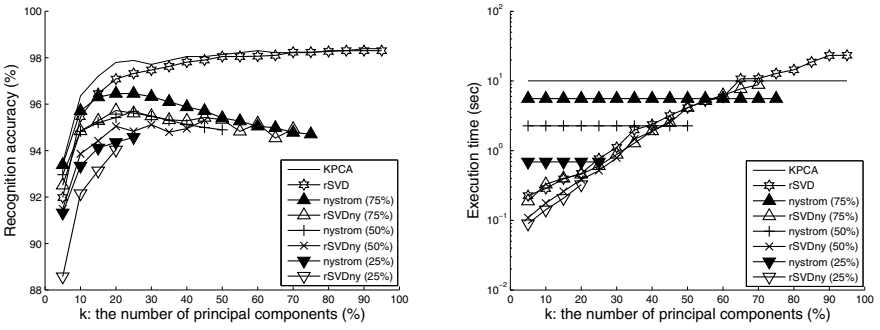
the result shows that the Nyström method with more samples tends to achieve better accuracy. However, the computation time of Nyström is proportional to m^3, so the system should select appropriate m in advance considering a trade-off between accuracy and time according to the size of the training set n.

For k, all Nyström methods show similar trend, although the original KPCA doesn't: each Nyström's accuracy increases until around $k = 25\%$, and then decreases. In our case, this number is 295 and it is equal to the number of class labels. Thus, the number of class labels can be a good candidate for selecting k.

3.4 Comparison with Nyström KPCA Ensemble

We compare the Nyström method with Nyström KPCA ensemble. In Nyström KPCA ensemble, we set $p = 150$ and $L = 2$. GCPs are randomly selected from the primal subset. After comparing execution time with the Nyström methods, we choose two different combinations of m and m_L: ENSEMBLE1=$\{m = 20\%, m_L = 20\%\}$, ENSEMBLE2=$\{m = 40\%, m_L = 30\%\}$. In the whole face recognition system, ENSEMBLE1 and ENSEMBLE2 take 0.96 and 2.02 seconds, where Nyström with 25%, 50%, and 75% sample size take 0.69, 2.27, and 5.58 seconds, respectively. (KPCA takes 10.05 seconds)

In Fig. 2 (b), Nyström KPCA ensemble achieves much better accuracy than the Nyström method with the almost same computation time. This is reasonable because ENSEMBLE1, or ENSEMBLE2, uses about three times more samples than Nyström with 25%, or 50%, sample size. The interesting thing is that ENSEMBLE1, which uses 60% of whole samples, gives better accuracy than even Nyström with 75% sample size.



(a) (b)

Fig. 3. (a) Face recognition accuracy and (b) execution time of KPCA, Nyström approximation, rSVD, and rSVDny (rSVD + Nyström) against variable m and k

3.5 Nyström vs. rSVD vs. Nyström + rSVD

We also compare the Nyström method with randomized SVD (rSVD) and rSVD + Nyström. Fig. 3 (a) shows that rSVD, or rSVD + Nyström, produces about 1% lower accuracy than KPCA, or Nyström, with same sample size. This performance decrease is caused after rSVD approximates the original eigendecomposition. In fact, there is a theoretical error bound for this approximation [1], so accuracy does not decrease significantly as you can see in the figure.

In Fig. 3 (b), as k increases, the computation time of rSVD and rSVD + Nyström increases exponentially, while that of Nyström remains same. At the end, rSVD even takes longer time than KPCA with large k. However, they still run as fast as Nyström with 25% sample size at $k = 25\%$, which is the best setting for XM2VTS database as we mentioned in section 3.3. Another interesting result is that the sample size m does not have much effect on the computation time of rSVD-based methods. This means that $O(mnk)$ from rSVD + Nyström and $O(n^2k)$ from rSVD are not much different when n is about 1180.

3.6 Experiments on Large-Scale Data

Now, we consider a large data set because our goal is to construct the large scale face recognition system. Previously, we used the simple classification method, correlation coefficient, but more complicated classification methods also can improve the classification accuracy. Thus, in this section, we compare the gram matrix reconstruction error, which is the standard measure for the Nyström method, rather than classification accuracy in order to leave room to apply different kind of classification methods. Because Nyström KPCA ensemble is not the gram matrix reconstruction method, its reconstruction errors are not as good as others, so we omit those results.

Since we only compare the gram matrix reconstruction error, we don't need the actual large scale face data. So we use Gisette data set from the UCI machine

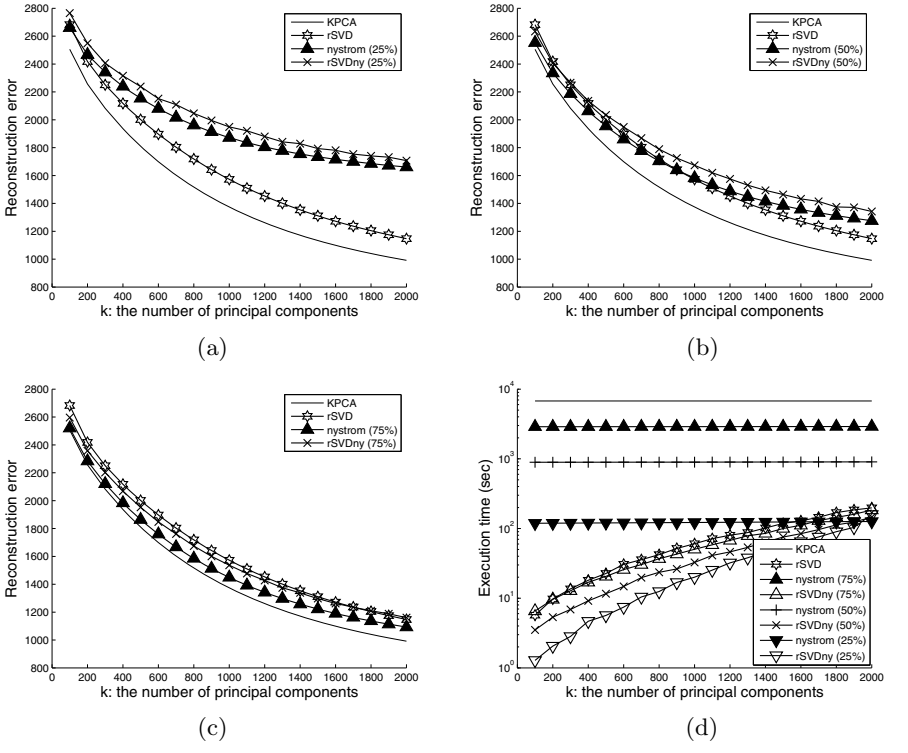


Fig. 4. (a)-(c) Gram matrix reconstruction error and (d) execution time of KPCA, Nyström approximation, rSVD, and rSVDny (rSVD + Nyström) against variable m and k for Gisette data

learning repository[1]. Gisette is a data set about handwritten digits of '4' and '9', which are highly confusable, and consists of 6,000 training set, 6,500 test set, and 1,000 validation set; each one is a collection of images at resolution 28×28. We compute the gram matrix of 12,500 images, training set + test set, using polynomial kernel $k(x, y) = \langle x, y \rangle^d$ with $d = 2$.

Similar to the previous experiment, rSVD, or rSVD + Nyström, shows same drop rate of the error compared to KPCA, or Nyström, with the slightly higher error (Fig. 4 (a)-(c)). As k increases, the Nyström method accumulates more error than KPCA, so we may infer that accuracy decreasing of Nyström in section 3.3 is caused by this accumulation. On the running time comparison (Fig. 4 (d)), same as the previous one (Fig. 3 (b)), the computation time of rSVD-based methods increases exponentially. But different from the previous, rSVD + Nyström terminates quite earlier than rSVD, which means the effect of m can be captured when $n = 12,500$.

[1] http://archive.ics.uci.edu/ml/datasets.html

4 Conclusions

In this paper we have considered a few methods for improving the scalability of SVD or KPCA, including Nyström approximation, Nyström KPCA ensemble, randomized SVD, and rSVD + Nyström, and have empirically compared them using face dataset and handwritten digit dataset. Experiments on face image dataset demonstrated that Nyström KPCA ensemble yielded better recognition accuracy than the standard Nyström approximation when both methods were applied in the same runtime environment. In general, rSVD or rSVD + Nyström was much faster but led to lower accuracy than Nyström approximation. Thus, rSVD + Nyström might be the method which provided a reasonable trade-off between speed and accuracy, as pointed out in [4].

Acknowledgments. This work was supported by the Converging Research Center Program funded by the Ministry of Education, Science, and Technology (No. 2011K000673), NIPA ITRC Support Program (NIPA-2011-C1090-1131-0009), and NRF World Class University Program (R31-10100).

References

1. Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. Arxiv preprint arXiv:0909.4061 (2009)
2. Kumar, S., Mohri, M., Talwalkar, A.: Sampling techniques for the Nyström method. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), Clearwater Beach, FL, pp. 304–311 (2009)
3. Lee, S., Choi, S.: Landmark MDS ensemble. Pattern Recognition 42(9), 2045–2053 (2009)
4. Li, M., Kwok, J.T., Lu, B.L.: Making large-scale Nyström approximation possible. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 631–638. Omnipress, Haifa (2010)
5. Messer, K., Matas, J., Kittler, J., Luettin, J., Maitre, G.: XM2VTSDB: The extended M2VTS database. In: Proceedings of the Second International Conference on Audio and Video-Based Biometric Person Authentification. Springer, New York (1999)
6. Schölkopf, B., Smola, A.J., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10(5), 1299–1319 (1998)
7. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3(1), 71–86 (1991)
8. Williams, C.K.I., Seeger, M.: Using the Nyström method to speed up kernel machines. In: Advances in Neural Information Processing Systems (NIPS), vol. 13, pp. 682–688. MIT Press (2001)
9. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Computing Surveys 35(4), 399–458 (2003)

A Robust Face Recognition through Statistical Learning of Local Features

Jeongin Seo and Hyeyoung Park*

School of Computer Science and Engineering
Kyungpook National University
Sangyuk-dong, Buk-gu, Daegu, 702-701, Korea
{lain,hypark}@knu.ac.kr
http://bclab.knu.ac.kr

Abstract. Among various signals that can be obtained from humans, facial image is one of the hottest topics in the field of pattern recognition and machine learning due to its diverse variations. In order to deal with the variations such as illuminations, expressions, poses, and occlusions, it is important to find a discriminative feature which can keep core information of original images as well as can be robust to the undesirable variations. In the present work, we try to develop a face recognition method which is robust to local variations through statistical learning of local features. Like conventional local approaches, the proposed method represents an image as a set of local feature descriptors. The local feature descriptors are then treated as a random samples, and we estimate the probability density of each local features representing each local area of facial images. In the classification stage, the estimated probability density is used for defining a weighted distance measure between two images. Through computational experiments on benchmark data sets, we show that the proposed method is more robust to local variations than the conventional methods using statistical features or local features.

Keywords: face recognition, local features, statistical feature extraction, statistical learning, SIFT, PCA, LDA.

1 Introduction

Face recognition is an active topic in the field of pattern recognition and machine learning[1].Though there have been a number of works on face recognition, it is still a challenging topic due to the highly nonlinear and unpredictable variations of facial images as shown in Fig 1. In order to deal with these variations efficiently, it is important to develop a robust feature extraction method that can keep the essential information and also can exclude the unnecessary variational information.

Statistical feature extraction methods such as PCA and LDA[2,3] can give efficient low dimensional features through learning the variational properties of

* Corresponding Author.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 335–341, 2011.
© Springer-Verlag Berlin Heidelberg 2011



Fig. 1. Variations of facial images; expression, illumination, and occlusions

data set. However, since the statistical approaches consider a sample image as a data point (i.e. a random vector) in the input space, it is difficult to handle local variations in image data. Especially, in the case of facial images, there are many types of face-specific occlusions by sun-glasses, scarfs, and so on. Therefore, for the facial data with occlusions, it is hard to expect the statistical approaches to give good performances.

To solve this problem, local feature extraction methods, such as Gabor filter and SIFT, has also been widely used for visual pattern recognition. By using local features, we can represent an image as a set of local patches and can attack the local variations more effectively. In addition, some local features such as SIFT are originally designed to have robustness to image variations such as scale and translations[4]. However, since most local feature extractor are previously determined at the developing stage, they cannot absorb the distributional variations of given data set.

In this paper, we propose a robust face recognition method which have a statistical learning process for local features. As the local feature extractor, we use SIFT which is known to show robust properties to local variations of facial images [7,8]. For every training image, we first extract SIFT features at a number of fixed locations so as to obtain a new training set composed of the SIFT feature descriptors. Using the training set, we estimate the probability density of the SIFT features at each local area of facial images. The estimated probability density is then used to calculate the weight of each features in measuring distance between images. By utilizing the obtained statistical information, we expect to get a more robust face recognition system to partial occlusions.

2 Representation of Facial Images Using SIFT

As a local feature extractor, we use SIFT (Scale Invariant Feature Transform) which is widely used for visual pattern recognition. It consists of two main stages of computation to generate the set of image features. First, we need to determine how to select interesting point from a whole image. We call the selected interesting pixel keypoint. Second, we need to define an appropriate descriptor for the selected keypoints so that it can represent meaningful local properties of given images. We call it keypoint descriptor. Each image is represented by the

set of keypoints with descriptors. In this section, we briefly explain the keypoint descriptor of SIFT and how to apply it for representing facial images.

SIFT [4] uses scale-space Difference-Of-Gaussian (DOG) to detect keypoints in images. For an input image, $I(x, y)$, the scale space is defined as a function, $L(x, y, \sigma)$ produced from the convolution of a variable-scale Gaussian $G(x, y, \sigma)$ with the input image. The DOG function is defined as follows:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{1}$$

where k represents multiplicative factor.

The local maxima and minima of $D(x, y, \sigma)$ are computed based on its eight neighbors in current image and nine neighbors in the scale above and below. In the original work, keypoints are selected based on the measures of their stability and the value of keypoint descriptors. Thus, the number of keypoints and location depends on each image. In case of face recognition, however, the original work has a problem that only a few number of keypoints are extracted due to the lack of textures of facial images. To solve this problem, Dreuw [6] have proposed to select keypoints at regular image grid points so as to give us a dense description of the image content, which is usually called Dense SIFT. We also use this approach in the proposed face recognition method.

Each keypoint extracted by SIFT method is represented as a descriptor that is a 128 dimensional vector composed of four part: locus (location in which the feature has been selected), scale (σ), orientation, and magnitude of gradient. The magnitude of gradient $m(x, y)$ and the orientation $\Theta(x, y)$ at each keypoint located at (x, y) are computed as follows:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \tag{2}$$

$$\Theta(x, y) = tan^{-1}\left\{\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right\} \tag{3}$$

In order to apply SIFT to facial image representation, we first fix the number of keypoints (M) and their locations on a regular grid. Since each keypoint is represented by its descriptor vector κ, a facial image I can be represented by a set of M descriptor vectors, such as

$$\mathrm{I} = \{\kappa_1, \kappa_2, ..., \kappa_{\mathrm{M}}\}. \tag{4}$$

Based on this representation, we propose a robust face recognition method through learning of probability distribution of descriptor vectors κ.

3 Face Recognition through Learning of Local Features

3.1 Statistical Learning of Local Features for Facial Images

As described in the above section, an image I can be represented by a fixed number (M) of keypoints κ_m ($m = 1, \ldots, M$). When the training set of facial

images are given as $\{\mathrm{I}^i\}_{i=1,\ldots,N}$, we can obtain M sets of keypoint descriptors, which can be written as

$$T_m = \{\kappa_m^i | \kappa_m^i \in \mathrm{I}^i, i = 1, \ldots, N\}, m = 1, \ldots, M. \tag{5}$$

The set T_m has keypoint descriptors at a specific location (i.e. mth location) of facial images obtained from all training images. Using the set T_m,we try to estimate the probability density of mth descriptor vectors κ_m. As a simple preliminary approach, we use the multivariate Gaussian model for 128-dimensional random vector. Thus, the probability density function of mth keypoint descriptor κ_m can be written by

$$p_m(\kappa) = \mathcal{G}(\kappa|\mu_m, \mathbf{\Sigma}_m) = \frac{1}{\sqrt{2\pi}^{128}} \frac{1}{\sqrt{|\mathbf{\Sigma}|}} \exp\left\{ -\frac{1}{2}(\kappa - \mu_m)^T \mathbf{\Sigma}^{-1}(\kappa - \mu_m) \right\}. \tag{6}$$

The two model parameters, the mean μ_m and the covariance $\mathbf{\Sigma}_m$, can be estimated by sample mean and sample covariance matrix of the training set T_m, respectively.

3.2 Weighted Distance Measure for Face Recognition

Using the estimated probability density function, we can calculate the probability that each descriptor is observed at a specific position of the prototype image of human frontal faces. When a test image given, its keypoint descroptors can have corresponding probability values, and we can use them to find the weight values of each descriptor for calculating the distance between training image and test image.

When a test image I^{tst} is given, we apply SIFT and obtain the set of keypoint descriptors for the test image such as

$$\mathrm{I}^{tst} = \{\kappa_1^{tst}, \kappa_2^{tst}, ..., \kappa_M^{tst}\}. \tag{7}$$

For each keypoint descriptor κ_m^{tst} $(m = 1, ..., M)$, we calculate the probability density $p_m(\kappa_m^{tst})$ and normalize it so as to obtain a weight value w_m for each keypoint descriptor κ_m^{tst}, which can be written as

$$w_m = \frac{p_m(\kappa_m^{tst})}{\sum_{n=1}^{M} p_n(\kappa_n^{tst})}. \tag{8}$$

Then the distance between the test image and a training image I^i can be calculated by using the equation;

$$d(\mathrm{I}^{tst}, \mathrm{I}^i) = \sum_{m=1}^{M} w_m d(\kappa_m^{tst}, \kappa_m^i). \tag{9}$$

where $d(\cdot, \cdot)$ denotes a well known distance measure such as L_1norm and L_2 norm.

Since w_m depends on the mth local patch of test image, which is represented by mth keypoint descriptor, the weight can be considered as the importance of the local patch in measuring the distance between training image and test images. When some occlusions occur, the local patches including occlusions are not likely to the usual patch shown in the training set, and thus the weight becomes small. Based on this consideration, we expect that the proposed measure can give more robust results to the local variations by excluding occluded part in the measurement.

4 Experimental Comparisons

4.1 Facial Image Database with Occlusions

In order to verify the robustness of the proposed method, we conducted computational experiments on AR database [9] with local variations. We compare the proposed method with the conventional local approaches [6] and the conventional statistical methods [2,3]. The AR database consists of over 3,200 color images of frontal faces from 126 individuals: 70 men and 56 women. There are 26 different images for each person. For each subject, these were recorded in two different sessions separated by two weeks delay. Each session consists of 13 images which has differences in facial expression, illumination and partial occlusion. In this experiment, we selected 100 individuals and used 13 images taken in the first session for each individual. Through preprocessing, we obtained manually aligned images with the location of eyes. After localization, faces were morphed and then resized to 88 by 64 pixels. Sample images from three subjects are shown in Fig. 2. As shown in the figure, the AR database has several examples with occlusions. In the first experiments, three non-occluded images (i.e., Fig. 2. (a), (c), and (g)) from each person were used for training, and other ten images for each person were used for testing.

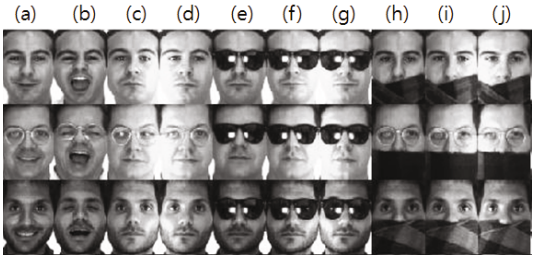


Fig. 2. Sample images of AR database

We also conducted additional experiments on the AR database with artificial occlusions. For each training image, we made ten test images by adding partial rectangular occlusions with random size and location to it. The generated sample images are shown in Fig. 3. These newly generated 3,000 images were used for testing.



Fig. 3. Sample images of AR database with artificial occlusions

4.2 Experimental Results

Using AR database, we compared the classification performance of the proposed method with a number of conventional methods: PCA, LDA, and dense SIFT with simple distance measure. For SIFT, we select a keypoint at every 16 pixels, so that we have 20 keypoint descriptor vectors for each image(i.e. M=20). For PCA, we take the eigenvectors so that the loss of information is less than 5%. For LDA, we use the feature set obtained through PCA for avoiding small sample set problem. After applying LDA, we use maximum dimension of feature vector which is limited to the number of classes. For classification, we used the nearest neighbor classifier with L_1 norm.

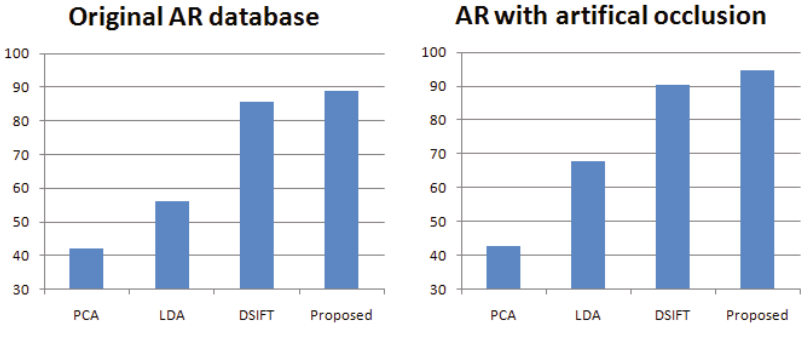


Fig. 4. Result of face recognition on AR database with occlusion

The result of the two experiments are shown in Fig. 4. In the first experiments on the original AR database, we can see that the statistical approaches give disappointing classification results. This may be due to the global properties of the statistical method, which is not appropriate for the images with local variations. Compared to statistical feature extraction method, we can see that the local features can give remarkably better results. In addition, by using the proposed weighted distance measure, the performance can be further improved. We can also see the similar results in the second experiments with artificial occlusions.

5 Conclusions

In this paper, we proposed a robust face recognition method by using statistical learning of local features. Through estimating the probability density of local features observed in training images, we can measure the importance of each local features of test images. This is a preliminary work on the statistical learning of local features using simple Gaussian model, and can be extended to more general probability density model and more sophisticated matching function. The proposed method can also be applied other types of visual recognition problems such as object recognition by choosing appropriate training set and probability density model of local features.

Acknowledgments. This research was partially supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0003671).

This research was partially supported by the Converging Research Center Program funded by the Ministry of Education, Science and Technology (2011K000659).

References

1. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Comput. Surv. 35(4), 399–458 (2003)
2. Martinez, A.M., Kak, A.C.: PCA versus LDA. IEEE Trans. Pattern Anal. Mach. Intell. 23(2), 228–233 (2001)
3. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of cognitive neuroscience 3(1), 71–86 (1991)
4. Lowe, D.G.: Distinctive image features from Scale-Invariant keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
5. Bicego, M., Lagorio, A., Grosso, E., Tistarelli, M.: On the use of SIFT features for face authentication. In: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop, vol. 35, IEEE Computer Society (2006)
6. Dreuw, P., Steingrube, P., Hanselmann, H., Ney, H., Aachen, G.: SURF-Face: face recognition under viewpoint consistency constraints. In: British Machine Vision Conference, London, UK (2009)
7. Cho, M., Park, H.: A Robust Keypoints Matching Strategy for SIFT: An Application to Face Recognition. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009. LNCS, vol. 5863, pp. 716–723. Springer, Heidelberg (2009)
8. Kim, D., Park, H.: An Efficient Face Recognition through Combining Local Features and Statistical Feature Extraction. In: Zhang, B.-T., Orgun, M.A. (eds.) PRICAI 2010. LNCS (LNAI), vol. 6230, pp. 456–466. Springer, Heidelberg (2010)
9. Martinez, A., Benavente, R.: The AR face database. CVC Technical Report #24 (June 1998)
10. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008)

Development of Visualizing Earphone and Hearing Glasses for Human Augmented Cognition

Byunghun Hwang[1], Cheol-Su Kim[1], Hyung-Min Park[2], Yun-Jung Lee[1], Min-Young Kim[1], and Minho Lee[1]

[1] School of Electronics Engineering, Kyungpook National University
{elecun,yjlee}@ee.knu.ac.kr, kcs6694@hanmail.net,
{minykim,mholee}@knu.ac.kr
[2] Department of Electronic Engineering, Sogang University
hpark@sogang.ac.kr

Abstract. In this paper, we propose a human augmented cognition system which is realized by a visualizing earphone and a hearing glasses. The visualizing earphone using two cameras and a headphone set in a pair of glasses intreprets both human's intention and outward visual surroundings, and translates visual information into an audio signal. The hearing glasses catch a sound signal such as human voices, and not only finds the direction of sound sources but also recognizes human speech signals. Then, it converts audio information into visual context and displays the converted visual information in a head mounted display device. The proposed two systems includes incremental feature extraction, object selection and sound localization based on selective attention, face, object and speech recogntion algorithms. The experimental results show that the developed systems can expand the limited capacity of human cognition such as memory, inference and decision.

Keywords: Computer interfaces, Augmented cognition system, Incremental feature extraction, Visualizing earphone, Hearing glasses.

1 Introduction

In recent years, many researches have been adopted the novel machine interface with real-time analysis of the signals from human neural reflexes such as EEG, EMG and even eye movement or pupil reaction, especially, for a person having a physical or mental condition that limits their senses or activities, and robot's applications. We already know that a completely paralyzed person often uses an eye tracking system to control a mouse cursor and virtual keyboard on the computer screen. Also, the handicapped are used to attempting to wear prosthetic arm or limb controlled by EMG. In robotic application areas, researchers are trying to control a robot remotely by using human's brain signals [2], [3].

Due to intrinsic restrictions in the number of mental tasks that a person can execute at one time, human cognition has its limitation and this capacity itself may fluctuate from moment to moment. As computational interfaces have become more prevalent nowadays and increasingly complex with regard to the volume and type of

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 342–349, 2011.
© Springer-Verlag Berlin Heidelberg 2011

information presented, many researchers are investigating novel ways to extend an information management capacity of individuals.

The applications of augmented cognition research are numerous, and of various types. Hardware and software manufacturers are always eager to employ technologies that make their systems easier to use, and augmented cognition systems would like to attribute to increase the productivity by saving time and money of the companies that purchase these systems. In addition, augmented cognition system technologies can also be utilized for educational settings and guarantee students a teaching strategy that is adapted to their style of learning. Furthermore, these technologies can be used to assist people who have cognitive or physical defects such as dementia or blindness. In one word, applications of augmented cognition can have big impact on society at large.

As we mentioned above, human brain has its limit to have attention at one time so that any kinds of augment cognition system will be helpful whether the user is disabled or not. In this paper, we describe our augmented cognition system which can assist in expanding the capacity of cognition. There are two types of our system named "visualizing earphone" and "hearing glasses". The visualizing earphone using two cameras and two mono-microphones interprets human intention and outward visual surroundings, also translates visual information into synthesized voice or alert sound signal. The hearing glasses work in opposite concepts to the visualizing earphone in the aspect of functional factors.

This paper is organized as follows. Section 2 depicts a framework of the implemented system. Section 3 presents experimental evaluation for our system. Finally, Section 4 summarizes and discusses the studies and future research direction.

2 Framework of the Implemented System

We developed two glasses-type's platforms to assist in expanding the capacity of human cognition, because of its convenience and easy-to-use. One is called "visualizing earphone" that has a function of translation from visual information to auditory information. The other is called "hearing glasses" that can decode auditory information into visual information. Figure 1 shows the implemented systems.

In case of visualizing earphone, in order to select one object which fits both interests and something salient, one of the cameras is mounted to the front side for capturing image of outward visual surroundings and the other is attached to the right side of the glasses for user's eye movement detection. In case of hearing glasses, mounted 2 mono-microphones are utilized to obtain the direction of sound source and to recognize speaker's voice. A head mounted display (HMD) device is used for displaying visual information which is translated from sound signal. Figure 2 shows the overall block diagram of the framework for visualizing earphone. Basically, hearing glasses functional blocks are not significantly different from this block diagram except the output manner. In this paper, voice recognition, voice synthesis and ontology parts will not discuss in detail since our work makes no contribution to those areas. Instead we focus our framework on incremental feature extraction method and face detection as well as recognition for augmented cognition.

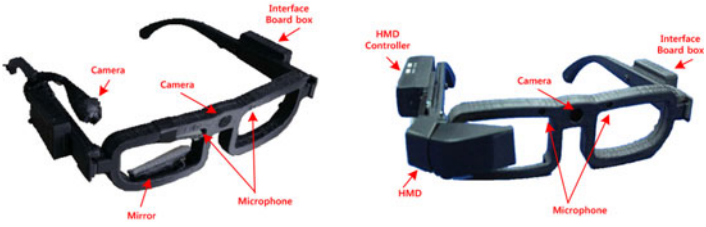


Fig. 1. "Visualizing earphone"(*left*) and "Hearing glasses"(*right*). Visualizing earphone has two cameras to find user's gazing point and small HMD device is mounted on the hearing glasses to display information translated from sound.

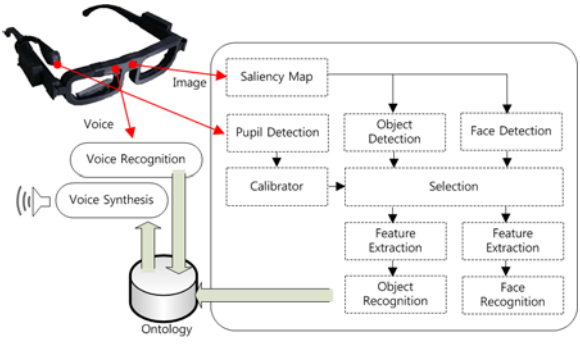


Fig. 2. Block diagram of the framework for the visualizing earphone

The framework has a variety of functionalities such as face detection using bottom-up saliency map, incremental face recognition using a novel incremental two dimensional two directional principle component analysis $(I(2D)^2PCA)$, gaze recognition, speech recognition using hidden Markov model(HMM) and information retrieval based on ontology, etc. The system can detect human intention by recognizing human gaze behavior, and it can process multimodal sensory information for incremental perception. In such a way, the framework will achieve the cognition augmentation.

2.1 Face Detection Based on Skin Color Preferable Selective Attention Model

For face detection, we consider skin color preferable selective attention model which is to localize a face candidate [11]. This face detection method has smaller computational time and lower false positive detection rate than well-known an Adaboost face detection algorithm. In order to robustly localize candidate regions for face, we make skin color intensified saliency map(SM) which is constructed by selective attention model reflecting skin color characteristics. Figure 3 shows the skin color preferable saliency map model.

A face color preferable saliency map is generated by integrating three different feature maps which are intensity, edge and color opponent feature map [1]. The face candidate regions are localized by applying a labeling based segmenting process. The

localized face candidate regions are subsequently categorized as final face candidates by the Haar-like form feature based Adaboost algorithm.

2.2 Incremental Two-Dimensional Two-Directional PCA

Reduction of computational load as well as memory occupation of a feature extraction algorithm is important issue in implementing a real time face recognition system. One of the most widespread feature extraction algorithms is principal component analysis which is usually used in the areas of pattern recognition and computer vision.[4] [5]. Most of the conventional PCAs, however, are kinds of batch type learning, which means that all of training samples should be prepared before testing process. Also, it is not easy to adapt a feature space for time varying and/or unseen data. If we need to add a new sample data, the conventional PCA needs to keep whole data to update the eigen vector. Hence, we proposed $(I(2D)^2PCA)$ to efficiently recognize human face [7].

After the $(2D)^2PCA$ is processed, the addition of a novel training sample may lead to change in both mean and covariance matrix. Mean is easily updated as follows,

$$\bar{x}' = \frac{1}{N+1}(N\bar{x} + y) \tag{1}$$

where y is a new training sample.

Changing the covariance means that eigenvector and eigenvalue are also changed. For updating the eigen space, we need to check whether an augment axis is necessary or not. In order to do, we modified accumulation ratio as in Eq. (2),

$$A'(k) = \frac{N(N+1)\sum_{i=1}^{k}\lambda_i + N \cdot tr([U_k^T(y-\bar{x})][U_k^T(y-\bar{x})]^T)}{N(N+1)\sum_{i=1}^{n}\lambda_i + N \cdot tr((y-\bar{x})(y-\bar{x})^T)} \tag{2}$$

where $tr(\bullet)$ is trace of matrix, N is number of training samples, λ_i is the i-th largest eigenvalue, $\bar{x}$ is a mean input vector, k and n are the number of dimensions of current feature space and input space, respectively.

We have to select one vector in residual vector set h, using following equation:

$$l = \arg\max A'([U, h_i]) \tag{3}$$

Residual vector set $h = [h_1, \cdots, h_n]$ is a candidate for a new axis. Based on Eq. (3), we can select the most appropriate axis which maximizes the accumulation ration in Eq. (2). Now we can find intermediate eigen problem as follows:

$$\left(\frac{N}{N+1}\begin{bmatrix}\Lambda & 0\\ 0^T & 0\end{bmatrix} + \frac{N}{(N+1)^2}\begin{bmatrix}gg^T & \gamma g\\ \gamma g^T & \gamma^2\end{bmatrix}\right)R = R\Lambda' \tag{4}$$

where $\gamma = h_l^T(y_l - \bar{x}_l)$, g is projected matrix onto eigen vector U, we can calculate the new $n \times (k+1)$ eigenvector matrix U' as follows:

$$U' = \left[U, \hat{h}\right]R \tag{5}$$

where

$$\hat{h} = \begin{cases} h_l / \|h_l\| & if\ A'(n) < \theta \\ 0 & otherwise \end{cases} \tag{6}$$

The I(2D)PCA only works for column direction. By applying same procedure to row direction for the training sample, I(2D)PCA is extended to I(2D)^{2}PCA.

2.3 Face Selection by Using Eye Movement Detection

Visualizing earphone should deliver the voice signals converted from visual data. At this time, if there are several objects or faces in the visual data, system should be able to select one among them. The most important thing is that the selected one should be intended by a user. For this reason, we adopted a technique which can track a pupil center in real time by using small IR camera with IR illuminations. In this case, we need to match pupil center position to corresponding point on the outside view image from outward camera. Figure 3 shows that how this system can select one of the candidates by using detection of pupil center after calibration process.

A simple second order polynomial transformation is used to obtain the mapping relationship between the pupil vector and the outside view image coordinate as shown in Eq. (7). Fitting even higher order polynomials has been shown to increase the accuracy of the system, but the second order requires less calibration points and provides a good approximation [8].

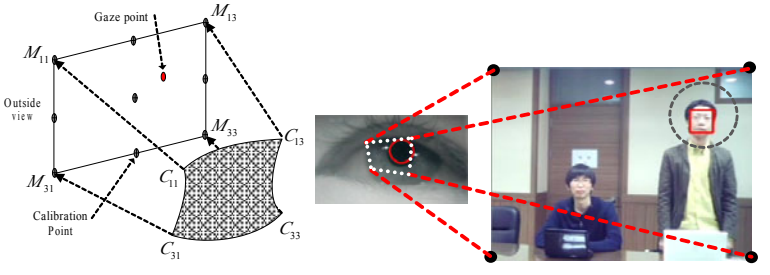


Fig. 3. Calibration procedure for mapping of coordinates between pupil center points and outside view points

$$\begin{aligned} \overline{x} &= a_0 x^2 + a_1 y^2 + a_2 x + a_3 y + a_4 xy + a_5 \\ \overline{y} &= b_0 x^2 + b_1 y^2 + b_2 x + b_3 y + b_4 xy + b_5 \end{aligned} \tag{7}$$

where $\overline{x}$ and $\overline{y}$ are the coordinates of a gaze point in the outside view image. Also, the parameters $a_0 \sim a_5$ and $b_0 \sim b_5$ in Eq. (7) are unknown. Since each calibration point can be represented by the $\overline{x}$ and $\overline{y}$ as shown in Eq. (7), the system has 12 unknown parameters but we have 18 equations obtained by the 9 calibration points for the $\overline{x}$ and $\overline{y}$ coordinates. The unknown parameters can be obtained by the least square algorithm. We can simply represent the Eq. (7) as the following matrix form.

$$M = TC \tag{8}$$

where M and C are the matrix represent the coordinates of the pupil and outside view image, respectively. T is a calibration matrix to be solved and play a mapping role between two coordinates. Thus, if we know the elements of M and C matrix, we can solve the calibration matrix T using M product inverse C matrix and then can obtain the matrix G which represents the gaze points correspond to the position of two eyes seeing the outside view image.

$$G = TW \tag{9}$$

where W is input matrix which represented the pupil center points.

2.4 Sound Localization and Voice Recognition

In order to select one of the recognized faces, besides a method using gaze point detection, sound localization based on histogram-based DUET (Degenerate Unmixing and Estimation Technique) [9] was applied to the system. Assuming that the time-frequency representation of the sources have disjoint support, the delay estimates obtained by relative phase differences between time-frequency segments from two-microphone signals may provide directions corresponding to source locations. After constructing a histogram by accumulating the delay estimates to achieve robustness, the direction corresponding to the peak of the histogram has shown a good performance for providing desired source directions under the adverse environments. Figure 4 shows the face selection strategy using sound localization.

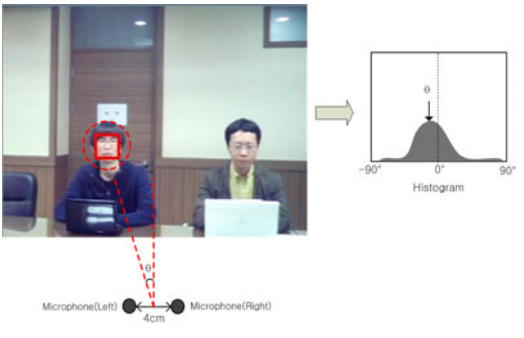


Fig. 4. Face selection by using sound localization

In addition, we employed the speaker independent speech recognition algorithm based on hidden Markov model [10] to the system for converting voice signals to visual signals. These methods are fused with the face recognition algorithm so the proposed augmented cognition system can provide more accurate information in spite of the noisy environments.

3 Experimental Evaluation

We integrated those techniques into an augmented cognition system. The system performance depends on the performance of each integrated algorithms. We experimentally evaluate the performance of entire system through the test for each algorithm.

In the face detection experiment, we captured 420 images from 14 videos for the training images to be used in each algorithm. We evaluated the performance of the face detection for UCD valid database (http://ee.ucd.ie/validdb/datasets.html). Even though the proposed model has slightly low true positive detection rate than that of the conventional Adaboost, but has better result for the false positive detection rate. The proposed model has 96.2% of true positive rate and 4.4% of false positive rate. Conventional Adaboost algorithm has 98.3% of true positive and 11.2% false positive rate.

We checked the performance of $I(2D)^2PCA$ by accuracy, number of coefficient and computational load. In test, proposed method is repeated by 20 times with different selection of training samples. Then, we used Yale database (http://cvc.yale.edu/projects/yalefaces/yalefa-ces.html) and ORL database (http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html) for the test. In case of using Yale data base, while incremental PCA has 78.47% of accuracy, the proposed algorithm has 81.39% of accuracy. With ORL database, conventional PCA has 84.75% of accuracy and proposed algorithm has 86.28% of accuracy. Also, the computation load is not much sensitive to the increasing number of training sample, but the computing load for the IPCA dramatically increase along with the increment number of sample data due to the increase of eigen axes.

In order to evaluate the performance of gaze detection, we divided the 800 x 600 screen into 7 x 5 sub-panels and demonstrated 10 times per sub-plane for calibration. After calibration, 12 target points are tested and each point is tested 10 times. The test result of gaze detection on the 800 x 600 resolution of screen. Root mean square error (RMSE) of the test is 38.489.

Also, the implemented sound localization system using histogram-based DUET processed two-microphone signals to record sound at a sampling rate of 16 kHz in real time. In a normal office room, localization results confirmed the system could accomplish very reliable localization under the noisy environments with low computational complexity.

Demonstration of the implemented human augmented system is shown in http://abr.knu.ac.kr/?mid=research.

4 Conclusion and Further Work

We developed two glasses-type platforms to expand the capacity of human cognition. Face detection using bottom up saliency map, face selection using eye movement detection, feature extraction using $I(2D)^2PCA$, and face recognition using Adaboost algorithm are integrated to the platforms. Specially, $I(2D)^2PCA$ algorithm was used to reduce the computational loads as well as memory size in feature extraction process and attributed to operate the platforms in real-time.

But there are some problems to be solved for the augmented cognition system. We should overcome the considerable challenges which have to provide correct information fitted in context and to process signals in real-world robustly, etc. Therefore, more advanced techniques such as speaker dependent voice recognition, sound localization and information retrieval system to interpret or understand the meaning of visual contents more accurately should be supported on the bottom. Therefore, we are attempting to develop a system integrated with these techniques.

Acknowledgments. This research was supported by the Converging Research Center Program funded by the Ministry of Education, Science and Technology (2011K000659).

References

1. Jeong, S., Ban, S.W., Lee, M.: Stereo saliency map considering affective factors and selective motion analysis in a dynamic environment. Neural Networks 21(10), 1420–1430 (2008)
2. Bell, C.J., Shenoy, P., Chalodhorn, R., Rao, R.P.N.: Control of a humanoid robot by a noninvasive brain-computer interface in humans. Journal of Neural Engineering, 214–220 (2008)
3. Bento, V.A., Cunha, J.P., Silva, F.M.: Towards a Human-Robot Interface Based on Electrical Activity of the Brain. In: IEEE-RAS International Conference on Humanoid Robots (2008)
4. Sirovich, L., Kirby, M.: Low-Dimensional Procedure for Characterization of Human Faces. J. Optical Soc. Am. 4, 519–524 (1987)
5. Kirby, M., Sirovich, L.: Application of the KL Procedure for the Characterization of Human Faces. IEEE Trans. on Pattern Analysis and Machine Intelligence 12(1), 103–108 (1990)
6. Lisin, D., Matter, M., Blaschko, M.: Combining local and global image features for object class recognition. IEEE Computer Vision and Pattern Recognition (2008)
7. Choi, Y., Tokumoto, T., Lee, M., Ozawa, S.: Incremental two-dimensional two-directional principal component analysis (I(2D)2PCA) for face recognition. In: International Conference on Acoustic, Speech and Signal Processing (2011)
8. Cherif, Z., Nait-Ali, A., Motsch, J., Krebs, M.: An adaptive calibration of an infrared light device used for gaze tracking. In: IEEE Instrumentation and Measurement Technology Conference, Anchorage, AK, pp. 1029–1033 (2002)
9. Rickard, S., Dietrich, F.: DOA estimation of many W-disjoint orthogonal sources from two mixtures using DUET. In: IEEE Signal Processing Workshop on Statistical Signal and Array Processing, pp. 311–314 (2000)
10. Rabiner, L.R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
11. Kim, B., Ban, S.-W., Lee, M.: Improving Adaboost Based Face Detection Using Face-Color Preferable Selective Attention. In: Fyfe, C., Kim, D., Lee, S.-Y., Yin, H. (eds.) IDEAL 2008. LNCS, vol. 5326, pp. 88–95. Springer, Heidelberg (2008)

Facial Image Analysis Using Subspace Segregation Based on Class Information

Minkook Cho and Hyeyoung Park*

School of Computer Science and Engineering,
Kyungpook National University, Daegu, South Korea
{mkcho,hypark}@knu.ac.kr

Abstract. Analysis and classification of facial images have been a challenging topic in the field of pattern recognition and computer vision. In order to get efficient features from raw facial images, a large number of feature extraction methods have been developed. Still, the necessity of more sophisticated feature extraction method has been increasing as the classification purposes of facial images are diversified. In this paper, we propose a method for segregating facial image space into two subspaces according to a given purpose of classification. From raw input data, we first find a subspace representing noise features which should be removed for widening class discrepancy. By segregating the noise subspace, we can obtain a residual subspace which includes essential information for the given classification task. We then apply some conventional feature extraction method such as PCA and ICA to the residual subspace so as to obtain some efficient features. Through computational experiments on various facial image classification tasks - individual identification, pose detection, and expression recognition - , we confirm that the proposed method can find an optimized subspace and features for each specific classification task.

Keywords: facial image analysis, principal component analysis, linear discriminant analysis, independant component analysis, subspace segregation, class information.

1 Introduction

As various applications of facial images have been actively developed, facial image analysis and classification have been one of the most popular topics in the field of pattern recognition and computer vision. An interesting point of the study on facial data is that a given single data set can be applied for various types of classification tasks. For a set of facial images obtained from a group of persons, someone needs to classify it according to the personal identity, whereas someone else may want to detect a specific pose of the face. In order to achieve

* Corresponding Author.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 350–357, 2011.
© Springer-Verlag Berlin Heidelberg 2011

good performances for the various problems, it is important to find a suitable set of features according to the given classification purposes.

The linear subspace methods such as PCA[11,7,8], ICA[5,13,3], and LDA[2,6,15] were successfully applied to extract features for face recognition. However, it has been argued that the linear subspace methods may fail in capturing intrinsic nonlinearity of data set with some environmental noisy variation such as pose, illumination, and expression. To solve the problem, a number of nonlinear subspace methods such as nonlinear PCA[4], kernel PCA[14], kernel ICA[12] and kernel LDA[14] have been developed. Though we can expect these nonlinear approaches to capture the intrinsic nonlinearity of a facial data set, we should also consider the computational complexity and practical tractability in real applications. In addition, it has been also shown that an appropriate decomposition of facespace, such as intra-personal space and extra-personal space, and a linear projection on the decomposed subspace can be a good alternative to the computationally difficult and intractable nonlinear method[10].

In this paper, we propose a novel linear analysis for extracting features for any given classification purpose of facial data. We first focus on the purpose of given classification task, and try to exclude the environmental noisy variation, which can be main cause of performance deterioration of the conventional linear subspace methods. As mentioned above, the environmental noise can be varied according to the purpose of tasks even for the same data set. For a same data set, a classification task is specified by the class label for each data. Using the data set and class label, we estimate the noise subspace and segregate it from original space. By segregating the noise subspace, we can obtain a residual space which include essential (hopefully intrinsically linear) features for the given classification task. For the obtained residual space, we extract low-dimensional features using conventional linear subspace methods such as PCA and ICA. In the following sections, we describe the proposed method in detail and experimental results with real facial data sets for various purposes.

2 Subspace Segregation

In this section, we describe overall process of the subspace segregation according to a given purpose of classification. Let us consider that we obtain several facial images from different persons with different poses. Using the given data set, we can conduct two different classification tasks: the face recognition and the pose detection. Even though the same data set is used for the two tasks, the essential information of the classification should be different according to the purpose. It means that the environmental noises are also different depending on the purpose. For example, the pose variation decreases the performance of face recognition task, and some personal features of individual faces decreases the performance of pose detection task. Therefore, it is natural to assume that original space can be decomposed into the noise subspace and the residual subspace. The features in the noise subspace caused by environmental interferences such as illumination often have undesirable effects on data resulting in the performance deterioration. If we can estimate the noise subspace and segregate it from the original

space, we can expect that the obtained residual subspace mainly has essential information such as class prototypes which can improve system performances for classification.

The goal of the proposed subspace segregation method is estimating the noise subspace which represents environmental variations within each class and eliminating that from the original space to decrease the varinace within a class and to increase the variance between classes. Fig. 1 shows the overall process of the proposed subspace segregation. We first estimate the noise subspace with the original data and then we project the original data onto the subspace in order to obtain the noise features in low dimensional subspace. After that, the low dimensional noise features are reconstructed in the original space. Finally, we can obtain the residual data by subtracting the reconstructed noise components from the original data.

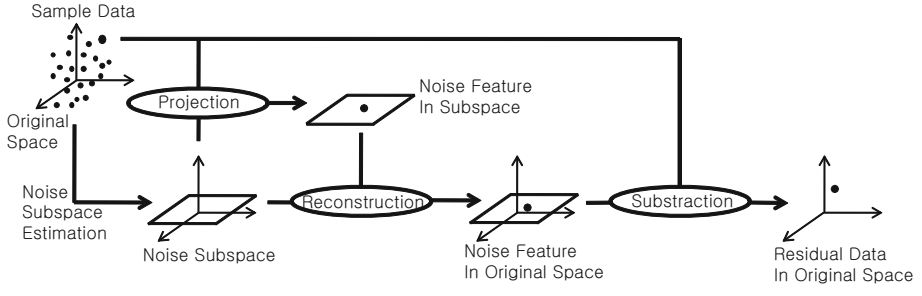


Fig. 1. Overall process of subspace segregation

3 Noise Subspace

For the subspace segregation, we first estimate the noise subspace from an original data. Since the noise features make the data points within a class be variant to each other, it consequently enlarges within-class variation. The residual features, which are obtained by eliminating the noise features, can be expected that it has some intrinsic information of each class with small variance.

To get the noise features, we first make a new data set defined by the difference vector $\boldsymbol{\delta}$ between two original data $\boldsymbol{x}_{ki}$, $\boldsymbol{x}_{kj}$ belonging to a same class $C_k (k = 1,...,K)$, which can be written as

$$\boldsymbol{\delta}_{kij} = \boldsymbol{x}_{ki} - \boldsymbol{x}_{kj}, \tag{1}$$

$$\Delta = \{\boldsymbol{\delta}_{kij}\}_{k=1,...,K, i=1,...,N_k, j=1,...,N_k}, \tag{2}$$

where $\boldsymbol{x}_{ki}$ denotes i-th data in class C_k and N_k denotes the number of data in class C_k. We can assume that Δ mainly represents within-class variations. Note that the set Δ is dependent on the class-label of data set. It implies that the obtained set Δ is defferent according to the classification purpose, even though the original data set is common. Figure 2 shows sample images of Δ for

two different classification purposes: (a) face recognition and (b) pose detection. From this figure, we can easily see that Δ of (a) mainly represents pose variation, and Δ of (b) mainly represents individual face variation.

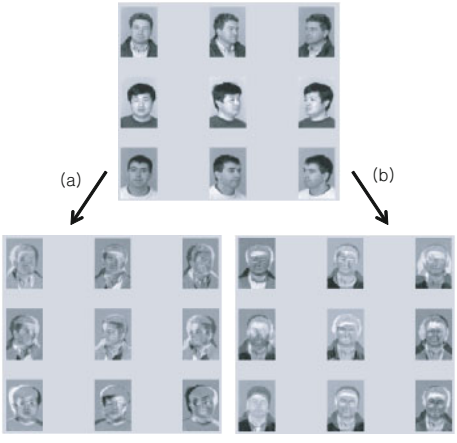


Fig. 2. The sample images of Δ; (a) face recognition and (b) pose detection

Since we want to find the dominant information of data set Δ, we apply PCA to Δ for obtaining the basis of the noise subspace such as

$$\Sigma_{\Delta} = V \Lambda V^T \tag{3}$$

where Σ_{Δ} is the covariance matrix and Λ are the eigenvalue matrix. Using the obtained basis of the noise subspace, the original data set X is projected to this subspace so as to get the low dimensional noise features(Y^{noise}) set through the calculation;

$$Y^{noise} = V^T X. \tag{4}$$

Since the obtained low dimensional noise feature is not desirable for classification, we need to eliminate it from the original data. To do this, we first reconstruct the noise components X^{noise} in original dimension from the low dimensional noise features Y^{noise} through the calculation;

$$X^{noise} = V Y^{noise} = V V^T X. \tag{5}$$

In the following section 4, we describe how to segregate X^{noise} from the original data.

4 Residual Subspace

Let us describe a definition of the residual subspace and how to get this in detail. Through the subspace segregation process, we obtain noise components

in original dimension. Since the noise features are not desirable for classification, we have to eliminate them from original data. To achieve this, we take the residual data X^{res} which can be computed by subtracting the noise features from the original data as follows

$$X^{res} = X - X^{noise} = (I - VV^T)X. \tag{6}$$

Figure 3 shows the sample images of the residual data for two different purposes: (a) face recognition and (b) pose detection. From this figure, we can see that 3-(a) is more suitable for face recognition than 3-(b), and vice versa. Using this residual data, we can expect to increase classification performance for the given purpose. As a further step, we apply a linear feature extraction method such as PCA and ICA, so as to obtain a residual subspace giving low dimensional features for the given classification task.

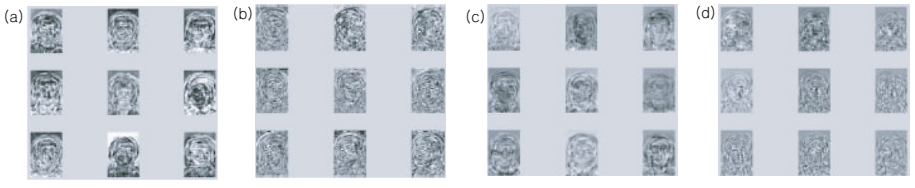


Fig. 3. The residual image samples (a, b) and the eigenface(c, d) for face recognition and pose detection, respectively

Figure 3-(c) and (d) show the eigenfaces obtained by applying PCA to the obtained residual data for face recognition and pose detection, respectively. Figure 3-(c) represents individual feature of each person and Figure 3-(d) represents some outlines of each pose. Though we only show the eigenfaces obtained by PCA, any other feature extraction can be applied. In the computational experiments in Section 5, we also apply ICA to obtain residual features.

5 Experiments

In order to confirm applicability of the proposed method, we conducted experiments on the real facial data sets and compared the performances with conventional methods.

We obtained some benchmark data sets from two different database: FERET (Face Recognition Technology) database and PICS(Psychological Image Collection at Stirling) database. From the FERET database at the homepage($http : //www.itl.nist.gov/iad/mumanid/feret/$), we selected 450 images from 50 persons. Each person has 9 images taken at 0°, ±15°, ±25°, ±40° and ±60° in viewpoint. We used this data set for face recognition as well as pose detection. From the PICS database at the homepage($http : //pics.psych.stir.ac.uk/$), we obtained 276 images from 69 persons. Each person has 4 images of different

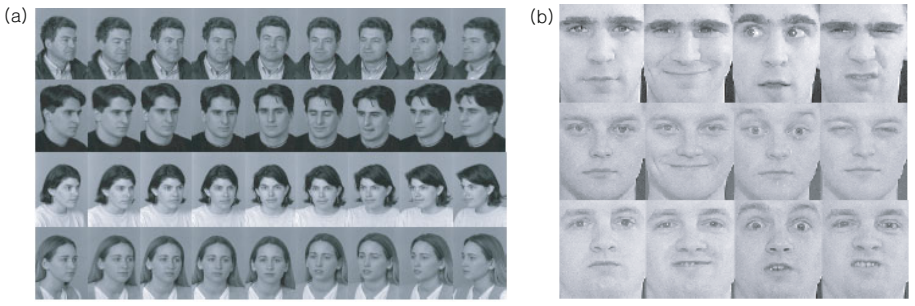


Fig. 4. The sample data from two databases; (a) FERET database and (b) PICS database

expressions. We used this data set for face recognition and facial expression recognition. Figure 4 shows the obtained sample data from two databases.

Face recognition task on the FERET database has 50 classes. In this class, three data images (left (+60°), right (-60°), and frontal (0°) images) are used for training, and the remaining 300 images were used for testing. For pose detection task, we have 9 classes with different viewpoints. For training, 25 data for each class were used, and the remaining 225 data were used for testing.

For facial expression recognition of PICS database, we have 4 classes(natural, happy, surprise, sad) For each class, 20 data were used for training and the remaining 49 data were used for testing. Finally, for face recognition we classified 69 classes. For training, 207 images (69 individuals, 3 images for each subject : sad, happy, surprise) were used and and the remaining 69 images were used for testing.

Table 1. Classification rates with FERET and PICS data

Database	Purpose	Origianl Data	Residual Data	PCA (dim)	LDA (dim)	Res. + ICA (dim)	Res. + PCA (dim)
FERET	Face Recognition	97.00	99.33	97.00 (117)	94.00 (30)	**100** (8)	**100** (8)
	Pose Detection	33.33	47.11	36.44 (65)	34.22 (8)	**58.22** (21)	**58.22** (21)
PICS	Expression Recognition	34.69	48.47	35.71 (65)	60.20 (3)	62.76 (32)	**66.33** (14)
	Face Recognition	72.46	88.41	72.46 (48)	57.97 (64)	**92.75** (89)	**92.75** (87)

In order to confirm plausibility of the residual data, we compared the performances on the original data with those the residual data. The nearest neighbor method[1,9] with Euclidean distance was adopted as a classifier. The experimental results are shown in Table 1. For the face recognition on FERET data, the

high performance can be achieved in spite of the large number of classes and limited number of training data, because the variations among classes are intrinsically high. On the other hand, the pose and facial expression recognition show generally low classification rates, due to the noise variations are extremely large and the class prototypes are terribly distorted by the noise. Nevertheless, the performance of the residual data shows better results than the original data in all the classification tasks.

We then apply some feature extraction methods to the residual data, and compared the performances with the conventional linear subspace methods. In Table 1, 'Res.' denotes the residual data and '(dim)' denotes the dimensionality of features. From the Table 1, we can confirm that the proposed methods using the residual data achieve significantly higher performances than the conventional PCA and LDA. For all classification tasks, the proposed methods of applying ICA or PCA give similar classification rates and the number of extracted features is also similar.

6 Conclusion

An efficient feature extraction method for various facial data classification problems was proposed. The proposed method starts from defining the "environmental noise" which is absolutely dependant on the purpose of given task. By estimating the noise subspace and segregating the noise components from the original data, we can obtain a residual subspace which includes essential information for the given classification purpose. Therefore, by just applying conventional linear subspace methods to the obtained residual space, we could achieve remarkable improvement in classification performance. Whereas many other facial analysis methods focus on the facial recognition problem, the proposed method can be efficiently applied to various analysis of facial data as shown in the computational experiments.

We should note that the proposed method is similar to the traditional LDA in the sense that the obtained residual features have small within-class variance. However, practical tractability of the proposed method is superior to LDA because it does not need to compute an inverse matrix of the within-scatter and the number of features does not depend on the number of classes. Though the proposed method adopts linear feature extraction methods, more sophisticated methods could possibly extract more efficient features from the residual space. In future works, the kernel methods or local linear methods could be applied to deal with non-linearity and complex distribution of the noise feature and the residual feature.

Acknowledgments. This research was partially supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2011-(C1090-1121-0002)).

This research was partially supported by the Converging Research Center Program funded by the Ministry of Education, Science and Technology (2011K000659).

References

1. Alpaydin, E.: Introduction to Machine Learning. The MIT Press (2004)
2. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 711–720 (1997)
3. Dagher, I., Nachar, R.: Face recognition using IPCA-ICA algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 996–1000 (2006)
4. DeMers, D., Cottrell, G.: Non-linear dimensionality reduction. In: Advances in Neural Information Processing Systems, pp. 580–580 (1993)
5. Draper, B.: Recognizing faces with PCA and ICA. Computer Vision and Image Understanding 91, 115–137 (2003)
6. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press (1990)
7. Mardia, K.V., Kent, J.T., Bibby, J.M.: Multivariate analysis. Academic Press (1979)
8. Martinez, A.M., Kak, A.C.: Pca versus lda. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 228–233 (2001)
9. Masip, D., Vitria, J.: Shared Feature Extraction for Nearest Neighbor Face Recognition. IEEE Transactions on Neural Networks 19, 586–595 (2008)
10. Moghaddam, B., Jebara, T., Pentland, A.: Bayesian face recognition. Pattern Recognition 33(11), 1771–1782 (2000)
11. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3, 71–86 (1991)
12. Yang, J., Gao, X., Zhang, D., Yang, J.: Kernel ICA: An alternative formulation and its application to face recognition. Pattern Recognition 38, 1784–1787 (2005)
13. Yang, J., Zhang, D., Yang, J.: Constructing PCA baseline algorithms to reevaluate ICA-based face-recognition performance. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 37, 1015–1021 (2007)
14. Yang, M.: Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods. In: IEEE International Conference on Automatic Face and Gesture Recognition, p. 215. IEEE Computer Society, Los Alamitos (2002)
15. Zhao, H., Yuen, P.: Incremental linear discriminant analysis for face recognition. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 38, 210–221 (2008)

An Online Human Activity Recognizer for Mobile Phones with Accelerometer

Yuki Maruno[1], Kenta Cho[2], Yuzo Okamoto[2], Hisao Setoguchi[2], and Kazushi Ikeda[1]

[1] Nara Institute of Science and Technology
Ikoma, Nara 630-0192 Japan
{yuki-ma,kazushi}@is.naist.jp
http://hawaii.naist.jp/
[2] Toshiba Corporation
Kawasaki, Kanagawa 212-8582 Japan
{kenta.cho,yuzo1.okamoto,hisao.setoguchi}@toshiba.co.jp

Abstract. We propose a novel human activity recognizer for an application for mobile phones. Since such applications should not consume too much electric power, our method should have not only high accuracy but also low electric power consumption by using just a single three-axis accelerometer. In feature extraction with the wavelet transform, we employ the Haar mother wavelet that allows low computational complexity. In addition, we reduce dimensions of features by using the singular value decomposition. In spite of the complexity reduction, we discriminate a user's status into walking, running, standing still and being in a moving train with an accuracy of over 90%.

Keywords: Context-awareness, Mobile phone, Accelerometer, Wavelet transform, Singular value decomposition.

1 Introduction

Human activity recognition plays an important role in the development of context-aware applications. If it is possible to have an application that determines a user's context such as walking or being in a moving train, the information can be used to provide flexible services to the user. For example, if a mobile phone with an application detects that the user is on a train, it can automatically switch to silent mode. Another possible application is to use the information for health care. If a mobile phone always records a user's status, the context will help a doctor give the user proper diagnosis.

Nowadays, mobile phones are commonly used in our lives and have enough computational power as well as sensors for applications with intelligent signal processing. In fact, they are utilized for human activity recognition as shown in the next section. In most of the related work, however, the sensors are multiple and/or fixed on a specific part of the user's body, which is not realistic for daily use in terms of electric power consumption of mobile phones or carrying styles.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 358–365, 2011.
© Springer-Verlag Berlin Heidelberg 2011

In this paper, we propose a human activity recognition method to overcome these problems. It is based on a single three-axis accelerometer, which is nowadays equipped to most mobile phones. The sensor does not need to be attached to the user's body in our method. This means the user can carry his/her mobile phone freely anywhere such as in a pocket or in his/her hands. For a direction-free analysis we perform preprocessing, which changes the three-axis data into device-direction-free data. Since the applications for mobile phones should not consume too much electric power, the method should have not only high accuracy but low power consumption. We use the wavelet transform, which is known to provide good features for discrimination [1]. To reduce the amount of computation, we use the Haar mother wavelet because the calculation cost is lower. Since a direct assessment from all wavelet coefficients will lead to large running costs, we reduce the number of dimensions by using the singular value decomposition (SVD). We discriminate the status into walking, running, standing still and being in a moving train with a neural network. The experimental results achieve over 90% of estimation accuracy with low power consumption.

The rest of this paper is organized as follows. In section 2, we describe the related work. In section 3, we introduce our proposed method. We show experimental result in section 4. Finally, we conclude our study in section 5.

2 Related Work

Recently, various sensors such as acceleration sensors and GPS have been mounted on mobile phones, which makes it possible to estimate user's activities with high accuracy. The high accuracy, however, depends on the use of several sensors and attachment to a specific part of the user's body, which is not realistic for daily use in terms of power consumption of mobile phones or carrying styles.

Cho et al. [2] estimate user's activities with a combination of acceleration sensors and GPS. They discriminate the user status into walking, running, standing still or being in a moving train. It is hard to identify standing still and being in a moving train. To tackle this problem, they use GPS to estimate the user's moving velocity. The identification of being in a moving train is easy with the user's moving velocity because the train moves at high speeds. Their experiments showed an accuracy of 90.6%, however, the problem is that the GPS does not work indoors or underground.

Mantyjarvi et al. [3] use two acceleration sensors, which are fixed on the user's hip. It is not really practical for daily use and their method is not suitable for the applications of mobile phones. The objective of their study is to recognize walking in a corridor, *Start/Stop point*, walking up and walking down. They combine the wavelet transform, principal component analysis and independent component analysis. Their experiments showed an accuracy of 83-90%.

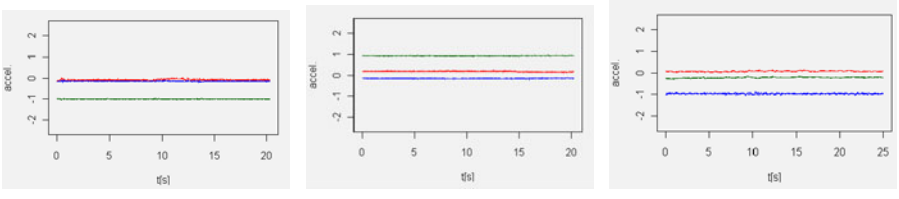
Iso et al. [1] propose a gait analyzer with an acceleration sensor on a mobile phone. They use wavelet packet decomposition for the feature extraction and classify them by combining a self-organizing algorithm with Bayesian

theory. Their experiments showed that their algorithm can identify gaits such as walking, running, going up/down stairs, and walking fast with an accuracy of about 80%.

3 Proposed Method

We discriminate a user's status into walking, running, standing still and being in a moving train based on a single three-axis accelerometer, which is equipped to mobile phones. Our proposed method works as follows.

1. Getting X, Y and Z-axis accelerations from a three-axis accelerometer (Fig.1).
2. Preprocessing for obtaining direction-free data (Fig.2).
3. Extracting the features using wavelet transform.
4. Selecting the features using singular value decomposition.
5. Estimating the user's activities with a neural network.



(a) standing still (b) standing still (c) train

Fig. 1. Example of "standing still" data and "train" data. These two "standing still" data differ from the position or direction of the sensor. "Train" data is similar to "standing still" data.

3.1 Preprocessing for Direction-Free Analysis

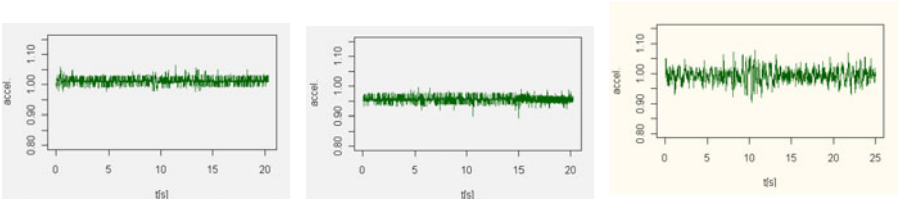
One of our goals is to adapt our method to applications for mobile phones. To realize this goal, the method does not depend on the position or direction of the sensor. Since the user carries a mobile phone with a three-axis accelerometer freely such as in a pocket or in his/her hands, we change the data (Fig.1) into device-direction-free data (Fig.2) by using Eq.(1).

$$\sqrt{X^2 + Y^2 + Z^2} \tag{1}$$

where X, Y and Z are the values of X, Y and Z-axis accelerations, respectively.

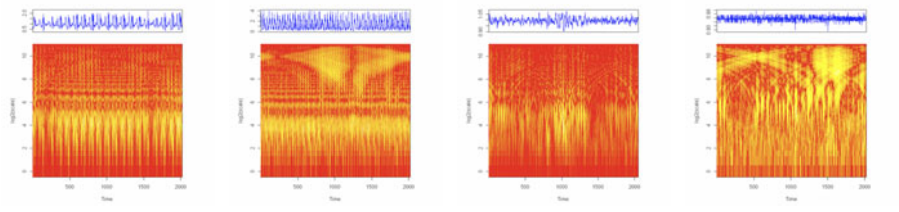
3.2 Extracting Features

A wavelet transform is used to extract the features of human activities from the preprocessed data. The wavelet transform is the inner-product of the wavelet



(a) standing still (b) standing still (c) train

Fig. 2. Example of preprocessed data. Original data is Fig.1.



(a) walking (b) running (c) standing still (d) being in a moving train

Fig. 3. Example of continuous wavelet transform

function with the signal $f(t)$. The continuous wavelet transform of a function $f(t)$ is defined as a convolution

$$W(a,b) = \langle f(t), \Psi_{a,b}(t) \rangle = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{a}} \Psi^*(\frac{t-b}{a}) dt \tag{2}$$

where $\Psi(t)$ is a continuous function in both the time domain and the frequency domain called the mother wavelet and the asterisk superscript denotes complex conjugation. The variables a(>0) and b are a scale and translation factor, respectively. $W(a,b)$ is the wavelet coefficient. Fig.3 is a plot of the wavelet coefficient. By using a wavelet transform, we can identify standing still and being in a moving train.

There are several mother wavelets such as Mexican hat mother wavelet (Eq.(3)) and Haar mother wavelet(Eq.(4)).

$$\Psi(t) = (1 - 2t^2)e^{-t^2} \tag{3}$$

$$\Psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \\ 0 & otherwise. \end{cases} \tag{4}$$

In our method, we use the Haar mother wavelet since it takes only two values and has a low computation cost. We evaluated the differences in the results for different mother wavelets. We compared the accuracy and calculation time

with Haar mother wavelet, Mexican hat mother wavelet and Gaussian mother wavelet. The experimental results showed that Haar mother wavelet is better.

3.3 Singular Value Decomposition

An application on a mobile phone should not consume too much electric power. Since a direct assessment from all wavelet coefficients would lead to large running costs, SVD of a wavelet coefficient matrix $\mathbf{X}$ is adopted to reduce the dimension of features.

A real (n × m) matrix, where n ≥ m $\mathbf{X}$ has the decomposition,

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathbf{T}} \tag{5}$$

where $\mathbf{U}$ is a n × m matrix with orthonormal columns ($\mathbf{U}^{\mathbf{T}}\mathbf{U} = \mathbf{I}$), while $\mathbf{V}$ is a m × m orthonormal matrix ($\mathbf{V}^{\mathbf{T}}\mathbf{V} = \mathbf{I}$) and $\boldsymbol{\Sigma}$ is a m × m diagonal matrix with positive or zero elements, called the singular values.

$$\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, ..., \sigma_m) \tag{6}$$

By convention it is assumed that $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_m \geq 0$.

3.4 Neural Network

We compared the accuracy and running time of two classifiers: neural networks (NNs), and support vector machines (SVMs). Since NNs are much faster than SVMs while their accuracies are comparable, we adopt an NN using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton method to classify human activities: walking, running, standing still, and being in a moving train. We use the largest singular value σ_1 of matrix Σ as an input value to discriminate the human activities.

4 Experiments

In order to verify the effectiveness of our method, we performed the following experiments. The objective of this study is to recognize walking, running, standing still, and being in a moving train. We used a three-axial accelerometer mounted on mobile phones. The testers carried their mobile phone freely such as in a pocket or in their hands. The data was logged with sampling rate of 100Hz. The data corresponding to being in a moving train was measured by one tester and the others were measured by seven testers in HASC2010corpus[1]. We performed the experiments on an Intel® XEON™ CPU 3.20GHz.

Table 1 shows the results. The accuracy rate was calculated against answer data.

[1] http://hasc.jp/hc2010/HASC2010corpus/hasc2010corpus-en.html

Table 1. The estimated accuracy. Sampling rate is 100Hz and time window is 1 sec.

	Walking	Running	Standing Still	Being in a train
Precision	93.5%	94.2%	92.7%	95.1%
Recall	96.0%	92.6%	93.6%	93.3%
F-measure	94.7%	93.4%	93.1%	94.2%

4.1 Running-Time Assessment

We aim at applying our method to mobile phones. For this purpose, the method should encompass high accuracy as well as low electric power consumption. We compared the accuracy with various sampling rates. We can save electric power consumption in the case of low sampling rate. Table 2 shows the results. As it can be seen, some of the results are below 90 %, however, as the time window becomes wider, the accuracy increases, which indicates that even if the sampling rate is low, we get better accuracy depending on the time window.

Table 2. The average accuracy for various sampling rates. The columns correspond to time windows of the wavelet transform.

	0.5s	1s	2s	3s
10Hz	84.9%	88.1%	90.7%	91.8%
25Hz	89.2%	92.6%	92.5%	92.5%
50Hz	90.5%	92.9%	94.1%	93.0%
100Hz	91.0%	93.9%	93.6%	93.6%

We compared our method with the previous method in terms of accuracy and computation time, where the input variables of the previous method are the maximum value and variance [2]. In Fig.4, our method in general showed higher accuracies. Although the previous method showed less computation time, the computation time of our method is enough for online processing (Fig.5).

4.2 Mother Wavelet Assessment

We also evaluated the differences in the results for different mother wavelets. We compared the accuracy and calculation time with Haar mother wavelet, Mexican hat mother wavelet and Gaussian mother wavelet. Table 3 and Table 4 show the accuracy for each mother wavelet and the calculation time per estimation, respectively. Although the accuracy is almost the same, the calculation time of Haar mother wavelet is much shorter than the others, which indicates that using Haar mother wavelet contributes to the reduction of electric power consumption.

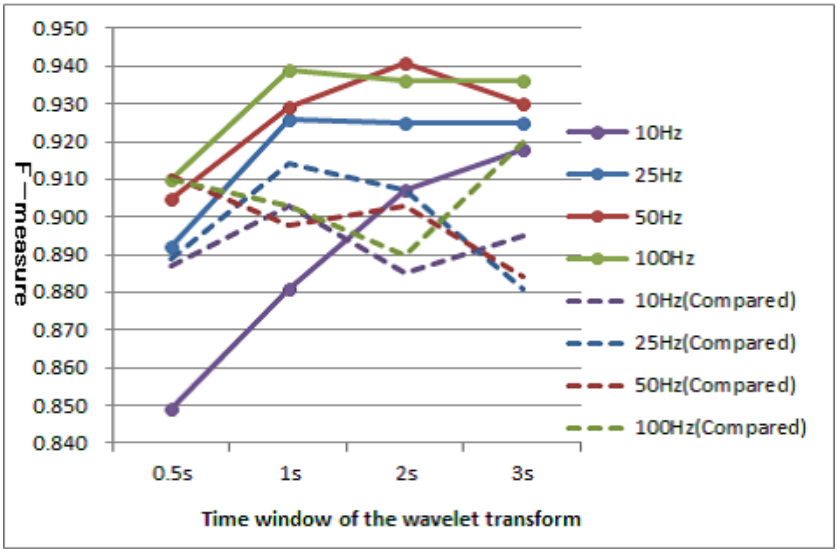


Fig. 4. The average accuracy for various sampling rates. Solid lines are our method while the ones in dash lines are previous compared method.

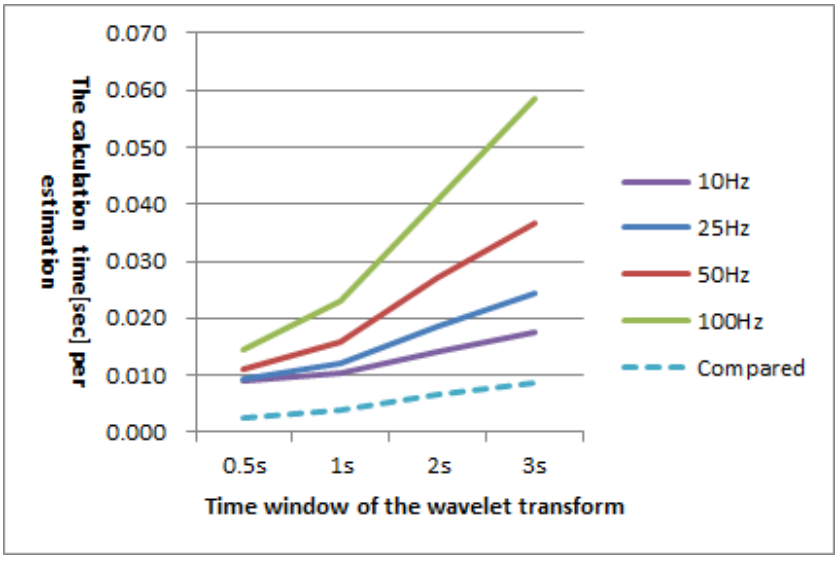


Fig. 5. The computation time per estimation for various sampling rates. Solid lines are our method while the one in dash line is previous compared method.

Table 3. The average accuracy for each Mother wavelet. The columns correspond to time windows of the wavelet transform.

	0.5s	1s	2s	3s
Haar	91.0%	93.9%	93.6%	93.6%
Mexican hat	91.1%	94.3%	93.9%	93.9%
Gaussian	91.2%	94.1%	93.5%	94.1%

Table 4. The calculation time[seconds] per estimation. The columns correspond to time windows of the wavelet transform.

	0.5s	1s	2s	3s
Haar	0.014sec	0.023sec	0.041sec	0.058sec
Mexican hat	0.029sec	0.062sec	0.129sec	0.202sec
Gaussian	0.029sec	0.061sec	0.128sec	0.200sec

5 Conclusion

We proposed a method that recognizes human activities using wavelet transform and SVD. Experiments show that freely positioned mobile phone equipped with an accelerometer could recognize human activities like walking, running, standing still, and being in a moving train with estimate accuracy of over 90% even in the case of low sampling rate. These results indicate that our proposed method can be successfully applied to commonly used mobile phones and is currently being implemented for commercial use in mobile phones.

References

1. Iso, T., Yamazaki, K.: Gait analyzer based on a cell phone with a single three-axis accelerometer. In: Proc. MobileHCI 2006, pp. 141–144 (2006)
2. Cho, K., Iketani, N., Setoguchi, H., Hattori, M.: Human Activity Recognizer for Mobile Devices with Multiple Sensors. In: Proc. ATC 2009, pp. 114–119 (2009)
3. Mantyjarvi, J., Himberg, J., Seppanen, T.: Recognizing human motion with multiple acceleration sensors. In: Proc. IEEE SMC 2001, vol. 2, pp. 747–752 (2001)
4. Daubechies, I.: The wavelet transform, time-frequency localization and signal analysis. In: Proc. IEEE Transactions on Information Theory, pp. 961–1005 (1990)
5. Le, T.P., Argou, P.: Continuous wavelet transform for modal identification using free decay response. Journal of Sound and Vibration 277, 73–100 (2004)
6. Kim, Y.Y., Kim, E.H.: Effectiveness of the continuous wavelet transform in the analysis of some dispersive elastic waves. Journal of the Acoustical Society of America 110, 86–94 (2001)
7. Shao, X., Pang, C., Su, Q.: A novel method to calculate the approximate derivative photoacoustic spectrum using continuous wavelet transform. Fresenius, J. Anal. Chem. 367, 525–529 (2000)
8. Struzik, Z., Siebes, A.: The Haar wavelet transform in the time series similarity paradigm. In: Proc. Principles Data Mining Knowl. Discovery, pp. 12–22 (1999)
9. Van Loan, C.F.: Generalizing the singular value decomposition. SIAM J. Numer. Anal. 13, 76–83 (1976)
10. Stewart, G.W.: On the early history of the singular value decomposition. SIAM Rev. 35(4), 551–566 (1993)

Preprocessing of Independent Vector Analysis Using Feed-Forward Network for Robust Speech Recognition

Myungwoo Oh and Hyung-Min Park

Department of Electronic Engineering, Sogang University,
#1 Shinsu-dong, Mapo-gu, Seoul 121-742, Republic of Korea

Abstract. This paper describes an algorithm to preprocess independent vector analysis (IVA) using feed-forward network for robust speech recognition. In the framework of IVA, a feed-forward network is able to be used as an separating system to accomplish successful separation of highly reverberated mixtures. For robust speech recognition, we make use of the cluster-based missing feature reconstruction based on log-spectral features of separated speech in the process of extracting mel-frequency cepstral coefficients. The algorithm identifies corrupted time-frequency segments with low signal-to-noise ratios calculated from the log-spectral features of the separated speech and observed noisy speech. The corrupted segments are filled by employing bounded estimation based on the possibly reliable log-spectral features and on the knowledge of the pre-trained log-spectral feature clusters. Experimental results demonstrate that the proposed method enhances recognition performance in noisy environments significantly.

Keywords: Robust speech recognition, Missing feature technique, Blind source separation, Independent vector analysis, Feed-forward network.

1 Introduction

Automatic speech recognition (ASR) requires noise robustness for practical applications because noisy environments seriously degrade performance of speech recognition systems. This degradation is mostly caused by difference between training and testing environments, so there have been many studies to compensate for the mismatch [1,2]. While recognition accuracy has been improved by approaches devised under some circumstances, they frequently cannot achieve high recognition accuracy for non-stationary noise sources or environments [3].

In order to simulate the human auditory system which can focus on desired speech even in very noisy environments, blind source separation (BSS) recovering source signals from their mixtures without knowing the mixing process has attracted considerable interest. Independent component analysis (ICA), which is the algorithm to find statistically independent sources by means of higher-order statistics, has been effectively employed for BSS [4]. As real-world acoustic

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 366–373, 2011.
© Springer-Verlag Berlin Heidelberg 2011

mixing involves convolution, ICA has generally been extended to the deconvolution of mixtures in both time and frequency domains. Although the frequency domain approach is usually favored due to high computational complexity and slow convergence of the time domain approach, one should resolve the permutation problem for successful separation [4]. While the frequency domain ICA approach assumes an independent prior of source signals at each frequency bin, independent vector analysis (IVA) is able to effectively improve the separation performance by introducing a plausible source prior that models inherent dependencies across frequency [5].

IVA employs the same structure as the frequency domain ICA approach to separate source signals from convolved mixtures by estimating an instantaneous separating matrix on each frequency bin. Since convolution in the time domain can be replaced with bin-wise multiplications in the frequency domain, these frequency domain approaches are attractive due to the simple separating system. However, the replacement is valid only when the frame length is long enough to cover the entire reverberation of the mixing process [6]. Unfortunately, acoustic reverberation is often too long in real-world situations, which results in unsuccessful source separation.

Kim *et al.* extended the conventional frequency domain ICA by using a feed-forward separating filter structure to separate source signals in highly reverberant conditions [6]. Moreover, this method adopted the minimum power distortionless response (MPDR) beamformer with extra null-forming constraints based on spatial information of the sources to avoid arbitrary permutation and scaling. A feed-forward separating filter network on each frequency bin was employed in the framework of the IVA to successfully separate highly reverberated mixtures with the exploitation of a plausible source prior that models inherent dependencies across frequency [7]. A learning algorithm for the network was derived with the extended non-holonomic constraint and the minimal distortion principle (MDP) [8] to avoid the inter-frame whitening effect and the scaling indeterminacy of the estimated source signals.

In this paper, we describe an algorithm that uses a missing feature technique to accomplish noise-robust ASR with preprocessing of the IVA using feed-forward separating filter networks. In order to discriminate reliable and unreliable time-frequency segments, we estimate signal-to-noise ratios (SNRs) from the log-spectral features of the separated speech and observed noisy speech and then compare them with a threshold. Among several missing feature techniques, we regard feature-vector imputation approaches since it may provide better performance by utilizing cepstral features and it does not have to alter the recognizer. In particular, the cluster-based reconstruction method is adopted since it can be more efficient than the covariance-based reconstruction method for a small training corpus by using a simpler model [9]. After filling unreliable time-frequency segments by the cluster-based reconstruction, the log-spectral features are transformed into cepstral features to extract MFCCs. Noise robustness of the proposed algorithm is demonstrated by speech recognition experiments.

2 Review on the IVA Using Feed-Forward Separating Filter Network

We briefly review the IVA method using feed-forward separating filter network [7] which is employed as a preprocessing step for robust speech recognition. Let us consider unknown sources, $\{s_i(t),\ i = 1, \cdots, N\}$, which are zero-mean and mutually independent. The sources are transmitted through acoustic channels and mixed to give observations, $x_i(t)$. Therefore, the mixtures are linear combinations of delayed and filtered versions of the sources. One of them can be given by

$$x_i(t) = \sum_{j=1}^{N} \sum_{p=0}^{L_m - 1} a_{ij}(p) s_j(t-p), \tag{1}$$

where $a_{ij}(p)$ and L_m denote a mixing filter coefficient and the filter length, respectively. The time domain mixtures are converted into frequency domain signals by the short-time Fourier transform, in which the mixtures can be expressed as

$$\mathbf{x}(\omega, \tau) = \mathbf{A}(\omega)\mathbf{s}(\omega, \tau), \tag{2}$$

where $\mathbf{x}(\omega, \tau) = [x_1(\omega, \tau)\ \cdots\ x_N(\omega, \tau)]^T$ and $\mathbf{s}(\omega, \tau) = [s_1(\omega, \tau)\ \cdots\ s_N(\omega, \tau)]^T$ denote the time-frequency representations of mixture and source signal vectors, respectively, at frequency bin ω and frame τ. $\mathbf{A}(\omega)$ represents a mixing matrix at frequency bin ω.

The source signals can be estimated from the mixtures by a network expressed as

$$\mathbf{u}(\omega, \tau) = \mathbf{W}(\omega)\mathbf{x}(\omega, \tau), \tag{3}$$

where $\mathbf{u}(\omega, \tau) = [u_1(\omega, \tau)\ \cdots\ u_N(\omega, \tau)]^T$ and $\mathbf{W}(\omega)$ denote the time-frequency representation of an estimated source signal vector and a separating matrix, respectively. If the conventional IVA is applied, the Kullback-Leibler divergence between an exact joint probability density function (pdf) $p(\mathbf{v}_1(\tau)\ \cdots\ \mathbf{v}_N(\tau))$ and the product of hypothesized pdf models of the estimated sources $\prod_{i=1}^{N} q(\mathbf{v}_i(\tau))$ is used to measure dependency between estimated source signals, where $\mathbf{v}_i(\tau) = [u_i(1, \tau)\ \cdots\ u_i(\Omega, \tau)]$ and Ω is the number of frequency bins [5]. After eliminating the terms independent of the separating network, the cost function is given by

$$J = -\sum_{\omega=1}^{\Omega} \log|\det \mathbf{W}(\omega)| - \sum_{i=1}^{N} E\{\log q(\mathbf{v}_i(\tau))\}. \tag{4}$$

The on-line natural gradient algorithm to minimize the cost function provides the conventional IVA learning rule expressed as

$$\Delta\mathbf{W}(\omega) \propto [\mathbf{I} - \varphi^{(\omega)}(\mathbf{v}(\tau))\mathbf{u}^H(\omega, \tau)]\mathbf{W}(\omega), \tag{5}$$

where the multivariate score function is given by $\varphi^{(\omega)}(\mathbf{v}(\tau)) = [\varphi^{(\omega)}(\mathbf{v}_1(\tau))\ \cdots\ \varphi^{(\omega)}(\mathbf{v}_N(\tau))]^T$ and $\varphi^{(\omega)}(\mathbf{v}_i(\tau)) = -\frac{\partial \log q(\mathbf{v}_i(\tau))}{\partial u_i(\omega, \tau)} = \frac{u_i(\omega, \tau)}{\sqrt{\sum_{\psi=1}^{\Omega} |u_i(\psi, \tau)|^2}}$. Desired time

domain source signals can be recovered by applying the inverse short-time Fourier transform to network output signals.

Unfortunately, since acoustic reverberation is often too long to express the mixtures with Eq. (2), the mixing and separating models should be extended to

$$\mathbf{x}(\omega,\tau)=\sum_{\kappa=0}^{K_m}\mathbf{A}'(\omega,\kappa)\mathbf{s}(\omega,\tau-\kappa), \tag{6}$$

and

$$\mathbf{u}(\omega,\tau)=\sum_{\kappa=0}^{K_s}\mathbf{W}'(\omega,\kappa)\mathbf{x}(\omega,\tau-\kappa), \tag{7}$$

where $\mathbf{A}'(\omega,\kappa)$ and K_m represent a mixing filter coefficient matrix and the filter length, respectively [6]. In addition, $\mathbf{W}'(\omega,\kappa)$ and K_s denote a separating filter coefficient matrix and the filter length, respectively. The update rule of the separating filter coefficient matrix based on minimizing the Kullback-Leibler divergence has been derived as

$$\begin{aligned}\Delta\mathbf{W}'(\omega,\kappa)\propto-\sum_{\mu=0}^{K_s}\{&\text{off-diag}(\varphi^{(\omega)}(\mathbf{v}(\tau-K_s))\mathbf{u}^H(\omega,\tau-K_s-\kappa+\mu))\\&+\beta(\mathbf{u}(\omega,\tau-K_s)-\mathbf{x}(\omega,\tau-3K_s/2))\mathbf{u}^H(\omega,\tau-K_s-\kappa+\mu)\}\mathbf{W}'(\omega,\mu),\end{aligned} \tag{8}$$

where 'off-diag$(\cdot)$' means a matrix with diagonal elements equal to zero and β is a small positive weighing constant [7]. In this derivation, non-causality was avoided by introducing a K_s-frame delay in the second term on the right side. In addition, the extended non-holonomic constraint and the MDP [8] were exploited to resolve scaling indeterminacy and whitening effect on the inter-frame correlations of estimated source signals. The feed-forward separating filter coefficients are initialized to zero, excluding the diagonal elements of $\mathbf{W}'(\omega,K_s/2)$ at all frequency bins which are initialized to one. To improve the performance, the MPDR beamformer with extra null-forming constraints based on spatial information of the sources can be applied before the separation processing [6].

3 Missing Feature Techniques for Robust Speech Recognition

Recovered speech signals obtained by the method mentioned in the previous section are exploited by missing feature techniques for robust speech recognition. The missing feature techniques is based on the observation that human listeners can perceive speech with considerable spectral excisions because of high redundancy of speech signals [10]. Missing feature techniques attempt either to obtain optimal decisions while ignoring time-frequency segments that are considered to be unreliable, or to fill in the values of those unreliable features. The cluster-based method to restore missing features was used, where the various spectral

profiles representing speech signals are assumed to be clustered into a set of prototypical spectra [10]. For each input frame, the cluster is estimated to which the incoming spectral features are most likely to belong from possibly reliable spectral components. Unreliable spectral components are estimated by bounded estimation based on the observed values of the reliable components and the knowledge of the spectral cluster to which the incoming speech is supposed to belong [10].

The original noisy speech and the separated speech signals are both used to extract log-spectral values in mel-frequency bands. Binary masks to discriminate reliable and unreliable log-spectral values for the cluster-based reconstruction method are obtained by [11]

$$M(\omega_{\mathrm{mel}}, \tau') = \begin{cases} 0, & L_{\mathrm{org}}(\omega_{\mathrm{mel}}, \tau') - L_{\mathrm{enh}}(\omega_{\mathrm{mel}}, \tau') \geq \mathrm{Th}, \\ 1, & \text{otherwise}, \end{cases} \tag{9}$$

where $M(\omega_{\mathrm{mel}}, \tau')$ denotes a mask value at mel-frequency band ω_{mel} and frame τ'. L_{org} and L_{enh} are the log-spectral values for the original noisy speech and the separated speech signals, respectively. The unreliable spectral components corresponding to zero mask values are reconstructed by the cluster-based method. The resulting spectral features are transformed into cepstral features, which are used as inputs of an ASR system [12].

4 Experiments

The proposed algorithm was evaluated through speech recognition experiments using the DARPA Resource Management database [13]. The training and test sets consisted of 3,990 and 300 sentences sampled at a rate of 16 kHz, respectively. The recognition system based on fully-continuous hidden Markov models (HMMs) was implemented by HMM toolkit [14]. Speech features were 13th-order mel-frequency cepstral coefficients with the corresponding delta and acceleration coefficients. The cepstral coefficients were obtained from 24 mel-frequency bands with a frame size of 25 ms and a frame shift of 10 ms.

The test set was generated by corrupting speech signal with babble noise [15]. Fig. 1 shows a virtual rectangular room to simulate acoustics from source positions to microphone positions. Two microphones were placed at positions marked by gray circles. The distance from a source to the center of two microphone positions was fixed to 1.5 m, and the target speech and babble noise sources were placed at azimuthal angles of $-20°$ and $50°$, respectively. To simulate observations at the microphones, target speech and babble noise signals were mixed with four room impulse responses from two speakers to two microphones which had been generated by the image method [16]. Since the original sampling rate (16 kHz) is too low to simulate signal delay at the two microphones close to each other, the source signals were upsampled to 1,024 kHz, convolved with room impulse responses generated at a sampling rate of 1,024 kHz, and downsampled back to 16 kHz. To apply IVA as a preprocessing step, the short-time Fourier transforms were conducted with a frame size of 128 ms and a frame shift of 32 ms.

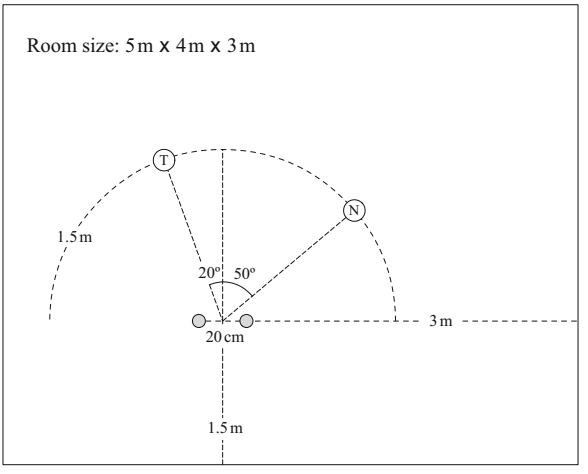


Fig. 1. Source and microphone positions to simulate corrupted speech

Table 1 shows the word accuracies in several echoic environments for corrupted speech signals whose SNR was 5 dB. As a preprocessing step, the conventional IVA method instead of the IVA using feed-forward network was also applied and compared in terms of the word accuracies. The optimal step size for each method was determined by extensive experiments. The proposed algorithm provided higher accuracies than the baseline without any processing for noisy speech and the method with the conventional IVA as a preprocessing step. For test speech signals whose SNR was varied from 5 dB to 20 dB, word accuracies accomplished by the proposed algorithm are summarized in Table 2. It is worthy

Table 1. Word accuracies in several echoic environments for corrupted speech signals whose SNR was 5 dB

Reverberation time	0.2 s	0.4 s
Baseline	24.9 %	16.4 %
Conventional IVA	75.1 %	29.7 %
Proposed method	80.6 %	32.2 %

Table 2. Word accuracies accomplished by the proposed algorithm for corrupted speech signals whose SNR was varied from 5 dB to 20 dB. The reverberation time was 0.2 s.

Input SNR	20 dB	15 dB	10 dB	5 dB
Baseline	88.0 %	75.2 %	50.8 %	24.9 %
Proposed method	90.6 %	88.4 %	84.9 %	80.6 %

of note that the proposed algorithm improved word accuracies significantly in these cases.

5 Concluding Remarks

In this paper, we have presented a method for robust speech recognition using cluster-based missing feature reconstruction with binary masks in time-frequency segments estimated by the preprocessing of IVA using feed-forward network. Based on the preprocessing which can efficiently separate target speech, robust speech recognition was achieved by identifying time-frequency segments dominated by noise in log-spectral feature domain and by filling the missing features with the cluster-based reconstruction technique. Noise robustness of the proposed algorithm was demonstrated by recognition experiments.

Acknowledgments. This research was supported by the Converging Research Center Program through the Converging Research Headquarter for Human, Cognition and Environment funded by the Ministry of Education, Science and Technology (2010K001130).

References

1. Juang, B.H.: Speech Recognition in Adverse Environments. Computer Speech & Language 5, 275–294 (1991)
2. Singh, R., Stern, R.M., Raj, B.: Model Compensation and Matched Condition Methods for Robust Speech Recognition. CRC Press (2002)
3. Raj, B., Parikh, V., Stern, R.M.: The Effects of Background Music on Speech Recognition Accuracy. In: IEEE ICASSP, pp. 851–854 (1997)
4. Hyvärinen, A., Harhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons (2001)
5. Kim, T., Attias, H.T., Lee, S.-Y., Lee, T.-W.: Blind Source Separation Exploiting Higher-Order Frequency Dependencies. IEEE Trans. Audio, Speech, and Language Processing 15, 70–79 (2007)
6. Kim, L.-H., Tashev, I., Acero, A.: Reverberated Speech Signal Separation Based on Regularized Subband Feedforward ICA and Instantaneous Direction of Arrival. In: IEEE ICASSP, pp. 2678–2681 (2010)
7. Oh, M., Park, H.-M.: Blind Source Separation Based on Independent Vector Analysis Using Feed-Forward Network. Neurocomputing (in press)
8. Matsuoka, K., Nakashima, S.: Minimal Distortion Principle for Blind Source Separation. In: International Workshop on ICA and BSS, pp. 722–727 (2001)
9. Raj, B., Seltzer, M.L., Stern, R.M.: Reconstruction of Missing Features for Robust Speech Recognition. Speech Comm. 43, 275–296 (2004)
10. Raj, B., Stern, R.M.: Missing-Feature Methods for Robust Automatic Speech Recognition. IEEE Signal Process. Mag. 22, 101–116 (2005)
11. Kim, M., Min, J.-S., Park, H.-M.: Robust Speech Recognition Using Missing Feature Theory and Target Speech Enhancement Based on Degenerate Unmixing and Estimation Technique. In: Proc. SPIE 8058 (2011), doi:10.1117/12.883340
12. Rabiner, L., Juang, B.-H.: Fundamentals of Speech Recognition. Prentice-Hall (1993)

13. Price, P., Fisher, W.M., Bernstein, J., Pallet, D.S.: The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition. In: Proc. IEEE ICASSP, pp. 651–654 (1988)
14. Young, S.J., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.C.: The HTK Book (for HTK Version 3.4). University of Cambridge (2006)
15. Varga, A., Steeneken, H.J.: Assessment for automatic speech recognition: II. In: NOISEX 1992: A Database and an Experiment to Study the Effect of Additive Noise on Speech Recognition Systems. Speech Comm., vol. 12, pp. 247–251 (1993)
16. Allen, J.B., Berkley, D.A.: Image Method for Efficiently Simulating Small-Room Acoustics. Journal of the Acoustical Society of America 65, 943–950 (1979)

Learning to Rank Documents Using Similarity Information between Objects

Di Zhou, Yuxin Ding, Qingzhen You, and Min Xiao

Intelligent Computing Research Center, Computer Science and Technology,
Harbin Institute of Technology Shenzhen Graduate School,
518055 Shenzhen, China
{zhoudi_hitsz,qzhyou,xiaomin_hitsz}@hotmail.com,
yxding@hitsz.edu.cn

Abstract. Most existing learning to rank methods only use content relevance of objects with respect to queries to rank objects. However, they ignore relationships among objects. In this paper, two types of relationships between objects, topic based similarity and word based similarity, are combined together to improve the performance of a ranking model. The two types of similarities are calculated using *LDA* and *tf-idf* methods, respectively. A novel ranking function is constructed based on the similarity information. Traditional gradient descent algorithm is used to train the ranking function. Experimental results prove that the proposed ranking function has better performance than the traditional ranking function and the ranking function only incorporating word based similarity between documents.

Keywords: learning to rank, lisewise, Latent Dirichlet Allocation.

1 Introduction

Ranking is widely used in many applications, such as document retrieval, search engine. However, it is very difficult to design effective ranking functions for different applications. A ranking function designed for one application often does not work well on other applications. This has led to interest in using machine learning methods for automatically learning ranked functions. In general, learning-to-rank algorithms can be categorized into three types, pointwise, pairwise, and listwise approaches.

The pointwise and pairwise approaches transform ranking problem into regression or classification on single object and object pairs respectively. Many methods have been proposed, such as Ranking SVM [1], RankBoost [2] and RankNet [3]. However, both pointwise and pairwise ignore the fact that ranking is a prediction task on a list of objects. Considering the fact, the listwise approach was proposed by Zhe Cao et. al [4]. In the listwise approach, a document list corresponding to a query is considered as an instance. The representative listwise ranking algorithms include ListMLE [5], ListNet[4], and RankCosine [6].

One problem of these listwise approaches mentioned above is that they only focus on the relationship between documents and queries, ignoring the similarity among documents. The relationship among objects when learning a ranking model is

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 374–381, 2011.
© Springer-Verlag Berlin Heidelberg 2011

considered in the algorithm proposed in paper [7]. But it is a pairwise ranking approach. One problem of pairwise ranking approaches is that the number of document pairs varies with the number of document [4], leading to a bias toward queries with more document pairs when training a model. Therefore, developing a ranking method with relationship among documents based on listwise approach is one of our targets.

To design ranking functions with relationship information among objects, one of the key problems we need to address is how to calculate the relationship among objects. The work [12] is our previous study on rank learning. In this paper each document is represented as a word vector, and the relationship between documents is calculated by the *cosine* similarity between two word vectors representing the two documents. We call this relationship as word relationship among objects. However, in practice when we say two documents are similar, usually we mean the two documents have similar topics. Therefore, in this paper we try to use topic similarity between documents to represent the relationship between documents. We call this relationship as topic relationship among objects.

The major contributions of this paper include (1) a novel ranking function is proposed for rank learning. This function not only considers content relevance of objects with respect to queries, but also incorporates two types of relationship information, word relationship among objects and topic relationship among objects. (2) We compare the performances of three types of ranking functions; they are the traditional ranking function, ranking function with word relationship among objects and the ranking function with word relationship and topic relationship among objects.

The remaining part of this paper is organized as follows. Section two introduces how to construct ranking function using word relationship information and topic relationship information. Section three discusses how to construct the loss functions for rank learning and gives the training algorithm to learn ranking function. Section four describes the experiment setting and experimental results. Section five is the conclusion.

2 Ranking Function with Topic Based Relationship Information

In this section, we discuss how to calculate topic relationships among documents and how to construct ranking function using relationships among documents.

2.1 Constructing Topic Relationship Matrix Based on LDA

Latent Dirichlet Allocation or LDA [8] was proposed by David M. Blei. LDA is a generation model and it can be looked as an approach that builds topic models using document clusters [9]. Compared to traditional methods, LDA can offer topic-level features corresponding to a document. In this paper we represent a document as a topic vectors, and then calculate the topic similarity between documents.

The architecture of LDA model is shown in Fig. 1. Assume that there are K topics and V words in a corpus. The corpus is a collection of M documents denoted as $D = \{d_1, d_2 \ldots d_M\}$. A document d_i is constructed by N words denoted as $\mathbf{w}_i = (w_{i1}, w_{i2} \ldots w_{iN})$. $\boldsymbol{\beta}$ is a $K \times V$ matrix, denoted as $\{\beta_k\}^K$. Each β_k denotes the mixture component

of topic k. $\boldsymbol{\theta}$ is a $M \times K$ matrix, denoted as $\{\theta_m\}^M$. Each θ_m denotes the topic mixture proportion for document d_m. In other words, each element $\theta_{m,k}$ of θ_m denotes the probability of document d_m belonging to topic k. We can obtain the probability for generating corpus D as following,

$$p(D|\alpha,\eta)=\prod_{d=1}^{M}\int p(\theta_d|\alpha)(\prod_{n=1}^{N_d}\sum_{z_{dn}} p(z_{dn}|\theta_d)p(w_{dn}|z_{dn},\eta))d\theta_d \tag{1}$$

where α denotes hyper parameter on the mixing proportions, η denotes hyper parameter on the mixture components, and z_{dn} indicates the topic for the nth word in document d.

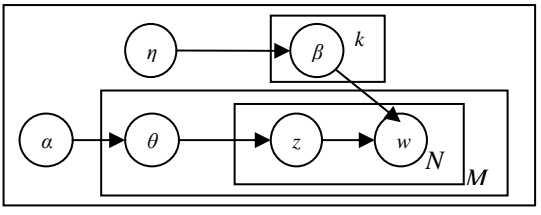


Fig. 1. Graphical model representation of LDA

In this paper, we utilize θ_m as the topic feature vectors of a document d_m, and the topic similarity between two documents is calculated by the *cosine* similarity of two topic vectors representing the two documents. We incorporate topic relationship and word relationship to calculate document rank. To calculate the word relationship, we represent document d_m as a word vector ζ_m. *tf-idf* method is employed to assign weights to words occurring in a document. The weight of a word is calculated according to (2).

$$w_{i,t}=\frac{TF_t(t,d_i)\log(\frac{n_i}{DF(t)})}{\sqrt{\sum_{t'\in V}TF_t^2(t',d_i)\log^2(\frac{n_i}{DF(t')})}} \tag{2}$$

In (2), $w_{i,t}$ indicates the weight assigned to term t. $TF_t(t, d_i)$ is the term frequency weight of term t in document d_i; n_i denotes the number of documents in the collection D_i, and $DF(t)$ is the number of documents in which term t occurs. The word similarity between two documents is calculated by the *cosine* similarity of two word vectors representing the two documents. In our experiments, we select the vocabulary by removing words in stop word list. This yielded a vocabulary of 2082 words in average.

The similarity measure defined in this paper incorporates topic similarity with word similarity, which is shown as (3). From (3) we can construct a M×M similarity matrix R to represent the relationship between objects, where $R^{(i,j)}$ and $R^{(j,i)}$ are equal to $sim(d_j, d_i)$. In our experiments, we set λ to 0.3 in ListMleNet and 0.5 in List2Net.

$$sim(d_m,d_{m'})=\lambda\cos(\theta_m,\theta_{m'})+(1-\lambda)\cos(\zeta_m,\zeta_{m'}),\quad 0<\lambda<1 \tag{3}$$

2.2 Ranking Function with Relationship Information among Objects

In this section we discuss how to design ranking function. Firstly, we define some notations used in this section. Let $Q = \{q_1, q_2, \ldots, q_n\}$ represent a given query set. Each query q_i is associated to a set of documents $D_i = \{d_{i1}, d_{i2}, \ldots, d_{im}\}$ where m denotes the number of documents in D_i. Each document d_{ij} in D_i is represented as a feature vector $\mathbf{x}_{ij} = \mathbf{\Phi}(q_i,d_{ij})$. The features in $\mathbf{x}_{ij}$ are defined in [10], which contain both conventional features (such as term frequency) and some ranking features (such as HostRank). Besides, each document set D_i is associated with a set of judgments $L_i = \{l_{i1}, l_{i2}, \ldots, l_{im}\}$, where l_{ij} is the relevance judgment of document d_{ij} with respect to query q_i. For example, l_{ij} can denote the position of document d_{ij} in ranking list, or represent the relevance judgment of document d_{ij} with respect to query q_i. R_i is the similarity matrix between documents in D_i. We can see each query q_i corresponds to a set of document D_i, a set of feature vectors $X_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2},\ldots, \mathbf{x}_{im}\}$, a set of judgments L_i [4], and a matrix R_i.

Let $f(X_i, R_i)$ denote a listwise ranking function for document set D_i with respect to query q_i . It outputs a ranking list for all documents in D_i. The ranking function for each document d_{ij} is defined as (4).

$$f(\mathbf{x}_{ij}, R_i \mid \zeta) = h(\mathbf{x}_{ij}, \mathbf{w}) + \tau \sum_{q \neq j}^{n_i} h(\mathbf{x}_{iq}, \mathbf{w}) \cdot R_i^{(j,q)} \cdot \tilde{R}_i^{(j,q)} \cdot \sigma(R_i^{(j,q)} \mid \zeta) \tag{4}$$

$$\sigma(R_i^{(j,q)} \mid \zeta) = \begin{cases} 1, & if\ R_i^{(j,q)} \geq \zeta \\ 0, & if\ R_i^{(j,q)} < \zeta \end{cases} \tag{5}$$

$$h(\mathbf{x}_{ij}, \mathbf{w}) = < \mathbf{x}_{ij}, \mathbf{w} > = \mathbf{x}_{ij} \cdot \mathbf{w} \tag{6}$$

where n_i denotes the number of documents in the collection D_i and feature vector $\mathbf{x}_{ij}$ denotes the content relevance of d_{ij} with respect to query q_i . $h(\mathbf{x}_{ij},\mathbf{w})$ in (6) is content relevance of d_{ij} with respect to query q_i .Vector $\mathbf{w}$ in $h(\mathbf{x}_{ij},\mathbf{w})$ is unknown, which is exactly what we want to learn. In this paper, $h(\mathbf{x}_{ij},\mathbf{w})$ is defined as a linear function, that is $h(.)$ takes inner product between vector $\mathbf{x}_{ij}$ and $\mathbf{w}$. $R_i^{(j,q)}$ denotes the similarity between document d_{ij} and d_{iq} as defined in (3). (5) is a threshold function. Its function is to prevent some documents which have little similarity with document d_{ij} affecting the rank of d_{ij} . ζ is constant, in our experiment set to 0.5. The second item of (4) can be interpreted as following: if the relevance score between d_{iq} and query q_i is high and d_{iq} is very similar with d_{ij} , then the relevance value between d_{ij} and q_i will be increased significantly, and vice versa. In (4) we can see the rank for document d_{ij} is decided by the content of d_{ij} and its similarities with other documents.

The coefficient τ is weight of similarity information (the second item of (4)). We can change its value to adjust the contribution of similarity information to the whole ranking value. In our experiment, we set it to 0.5. $\tilde{R}_i^{(j,q)}$ is a normalized value of $R_i^{(j,q)}$, which is calculated according to (7). Its function is to reduce the bias introduced by $R_i^{(j,q)}$. From (4) we can see that the ranking function (4) tends to give high rank to an object which has more similar documents without the normalized $\tilde{R}_i^{(j,q)}$. In [12] we analyzed this bias in detail.

$$\tilde{R}_i^{(j,q)} = \frac{R_i^{(j,q)}}{\sum_{r \neq j} R_i^{(j,r)}} \tag{7}$$

3 Training Algorithm of Ranking Function

In this section, we use two training algorithms to learn the proposed listwise rankings function. The two algorithms are called ListMleNet and List2Net, respectively. The only difference between the two algorithms is that they use different loss functions. ListMleNet uses the likelihood loss proposed by [5], and List2Net uses the cross entropy proposed by [4]. The two algorithms all use stochastic gradient descent algorithm to search the local minimum of loss functions. The stochastic gradient descent algorithm is described as Algorithm 1.

Table 1. Stochastic Gradient Descent Algorithm

Algorithm 1 Stochastic Gradient Descent
Input: training data $\{\{X_1, L_1, R_1\}, \{X_2, L_2, R_2\}, \ldots, \{X_n, L_n, R_n\}\}$
Parameter: learning rate η, number of iterations T
Initialize parameter $\mathbf{w}$
For $t = 1$ **to** T **do**
For $i = 1$ **to** n **do**
Input $\{X_i, L_i, R_i\}$ to Neural Network
Compute the gradient $\triangle \mathbf{w}$ with current $\mathbf{w}$
Update $\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{\eta} \times \frac{\partial \boldsymbol{L}(\boldsymbol{f}(\boldsymbol{X_i}, \boldsymbol{R_i})_{\boldsymbol{W}} \boldsymbol{L_i})}{\partial_{\boldsymbol{w}}}$
End for
End for
Output: $\mathbf{w}$

In table 1, the function $L(f(X_i,R_i)_{\mathbf{w}},L_i)$ denotes the surrogate loss function. In ListMleNet, the gradient of the likelihood loss $L(f(X_i,R_i)_{\mathbf{w}},L_i)$ with respect to $\mathbf{w}_j$ can be derived as (8). In List2Net the gradient of the cross entropy loss $L(f(X_i,R_i)_{\mathbf{w}},L_i)$ with respect to $\mathbf{w}_j$ can be derived as (9).

$$\begin{aligned} \Delta \mathbf{w}_j &= \frac{\partial \mathrm{L}(f(X_i,R_i)_{\mathbf{w}}, L_i)}{\partial \mathbf{w}_j} \\ &= -\frac{1}{\ln 10} \sum_{k=1}^{n_i} \{ \frac{\partial f(\mathbf{x}_{iL_i^k}, R_i)}{\partial \mathbf{w}_j} - \frac{\sum_{p=k}^{n_i} [\exp(f(\mathbf{x}_{iL_i^p}, R_i)) \cdot \frac{\partial f(\mathbf{x}_{iL_i^p}, R_i)}{\partial \mathbf{w}_j}]}{\sum_{p=k}^{n_i} \exp(f(\mathbf{x}_{iL_i^p}, R_i))} \} \end{aligned} \tag{8}$$

$$\Delta \mathbf{w}_j = \frac{\partial \mathrm{L}(f(X_i, R_i)_{\mathbf{w}}, L_i)}{\partial \mathbf{w}_j}$$

$$= -\sum_{k=1}^{n_i} [P_{L_i}(\mathbf{x}_{ik}) \cdot \frac{\partial f(\mathbf{x}_{ik}, R_i)}{\partial \mathbf{w}_j}] + \frac{\sum_{k=1}^{n_i} [\exp(f(\mathbf{x}_{ik}, R_i)) \cdot \frac{\partial f(\mathbf{x}_{ik}, R_i)}{\partial \mathbf{w}_j}]}{\sum_{k=1}^{n_i} \exp(f(\mathbf{x}_{ik}, R_i))} \quad (9)$$

In (8) and (9), $\frac{\partial f(\mathbf{x}_{ik}, R_i)}{\partial \mathbf{w}_j} = \mathbf{x}_{ik}^{(j)} + \tau \sum_{p=1, p \neq k} \mathbf{x}_{ip}^{(j)} R_i^{(k,p)} \tilde{R}_i^{(k,p)} \sigma(R_i^{(k,p)} \mid \zeta)$ and $\mathbf{x}_{ik}^{(j)}$ is the *j-th* element in $\mathbf{x}_{ik}$.

4 Experiments

We employed the dataset LETOR [10] to evaluate the performance of different ranking functions. The dataset contains 106 document collections corresponding to 106 queries. Five queries (8, 28, 49, 86, 93) and their corresponding document collections are discarded due to having no highly relevant query document pairs. In LETOR each document d_{ij} has been represented as a vector $\mathbf{x}_{ij}$. The similarity matrix R_i for *i*th query is calculated according to (3). We partitioned the dataset into *five* subsets and conducted 5-fold cross-validation. Each subset contains about 20 document collections. For performance evaluation, we adopted the IR evaluation measures: NDCG (Normalized Discounted Cumulative Gain) [11]. In the experiments we randomly selected one perfect ranking among the possible perfect rankings for each query as the ground truth ranking list.

In order to prove the effectiveness of the algorithm proposed in this paper, we compared the proposed algorithms with other two kind of listwise algorithms, ListMle[5] and listNet[4]. The difference of these algorithms is that they use different types of ranking functions and loss functions. In these algorithms two types of loss functions are used. They are likelihood loss (denoted as LL) and cross entropy (denoted as CE). In this paper we divide a ranking function into three parts. They are query relationship (denoted as QR), word relationship (denoted as WR) and topic relationship (denoted as TR). Query relationship refers to the content relevance of objects with respect to queries, that is the function $h(\mathbf{x}_{ij}, \mathbf{w})$ in (4). Word relationship and topic relationship have the same expression as the second term in (4). The difference between them is that word relationship uses the word similarity matrix (the first term in (3)), and topic relationship uses the topic similarity matrix (the second term in (3)). The performance comparison of different ranking learning algorithms is shown in Fig.2 and Fig.3, respectively. In Fig.2 and Fig.3, the x-axes represents top *n* documents; the y-axes is the value of NDCG; "TR n" represents n topics are selected by LDA.

ListMle and ListMleNet all use likelihood loss function. From Figure 2, we can get the following results: 1) ListMleNet (QR+WR) and ListMleNet (QR+WR+TR) outperform ListMle in terms of NDCG measures. In average the NDCG value of ListMleNet is about 1-2 points higher than ListMle. 2) The performance of

ListMleNet (QR+WR+TR) is affected by the topic numbers selected in LDA. In our experiments ListMleNet gets the best performance when topic number is 100. In average the NDCG value of ListMleNet (QR+WR+TR100) is about 0.3 points higher than ListMle(QR+WR). Especially, on NDCG@1 ListMleNet (QR+WR+TR100) has 2-point gain over ListMleNet (QR+WR). Therefore, topic similarity between documents is helpful for ranking documents.

ListNet and List2Net all use likelihood loss function. Their performances are shown in Fig.3. From Fig. 3, we can get the similar results: 1) List2Net (QR+WR) and List2Net (QR+WR+TR) outperform ListNet in terms of NDCG measures. In average the NDCG value of List2Net is about 1-2 points higher than ListNet. 2) The performance of List2Net (QR+WR+TR) is also affected by the topic numbers. In our experiments List2Net gets the best performance when topic number is 100. In average the NDCG value of List2Net (QR+WR+TR100) is about 0.9 points higher than ListNet (QR+WR). It is also shown that topic similarity between documents is helpful for ranking documents.

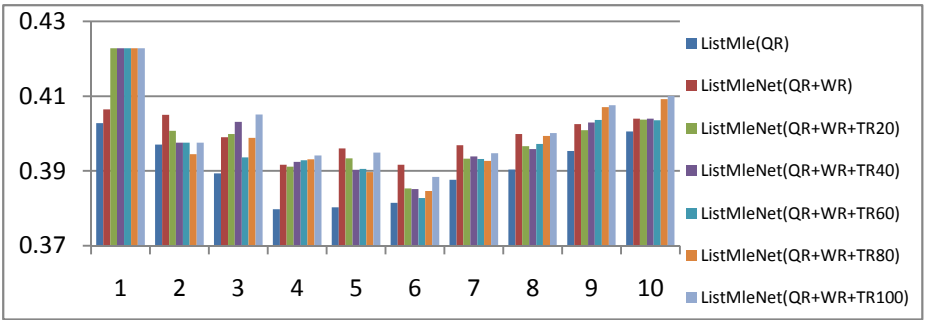


Fig. 2. Ranking performances of ListMle and ListMleNet

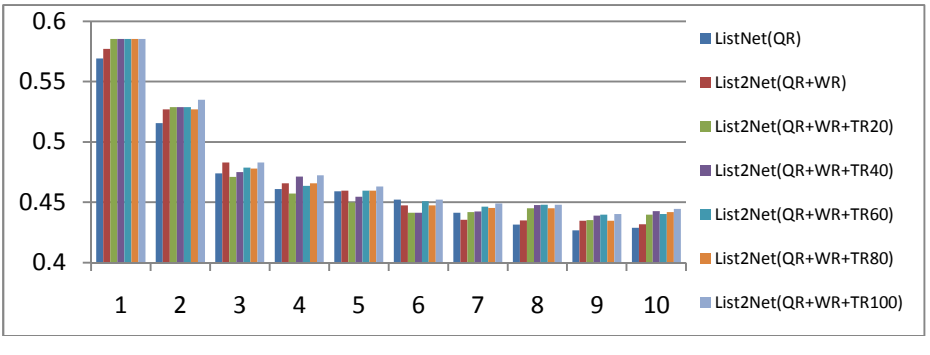


Fig. 3. Ranking performances of ListNet and List2Net

5 Conclusions

In this paper we use relationship information among objects to improve the performance of ranking model. Two types of relationship information, word

relationship and topic relationship among objects are incorporated into ranking function. Stochastic gradient descent algorithm is employed to learn ranking functions. Our experiments prove that ranking function with similarity information between objects performs better than the traditional ranking function and ranking functions with topic-based similarity information works more effectively than that only using word-based similarity information.

Acknowledgments. This work was partially supported by Scientific Research Foundation in Shenzhen (Grant No. JC201005260159A), Scientific Research Innovation Foundation in Harbin Institute of Technology (Project No. HIT.NSRIF2010123), and Key Laboratory of Network Oriented Intelligent Computation (Shenzhen).

References

1. Herbrich, R., Graepel, T., Obermayer, K.: Support vector learning for ordinal regression. In: Ninth International Conference on Artificial Neural Networks, pp. 97–102. ENNS Press, Edinburgh (1999)
2. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for Combining Preferences. Journal of Machine Learning Research 4, 933–969 (2003)
3. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: 22nd International Conference on Machine learning, pp. 89–96. ACM Press, New York (2005)
4. Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F., Li, H.: Learning to rank: from pairwise approach to listwise approach. In: 24th International Conference on Machine learning, pp. 129–136. ACM Press, New York (2007)
5. Xia, F., Liu, T.Y., Wang, J., Zhang, W., Li, H.: Listwise Approach to Learning to Rank: Theory and Algorithm. In: 25th International Conference on Machine Learning, pp. 1192–1199. ACM Press, New York (2008)
6. Qin, T., Zhang, X.D., Tsai, M.F., Wang, D.S., Liu, T.Y., Li, H.: Query-level loss functions for information retrieval. Information Processing and Management 44, 838–855 (2008)
7. Qin, T., Liu, T.Y., Zhang, X.D., Wang, D.S., Xiong, W.Y., Li, H.: Learning to Rank Relational Objects and Its Application to Web Search. In: 17th International World Wide Web Conference Committee, pp. 407–416. ACM Press, New York (2008)
8. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
9. Wei, X., Croft, W.B.: LDA-Based Document Models for Ad-hoc Retrieval. In: 29th SIGIR Conference, pp. 178–185. ACM Press, New York (2006)
10. Liu, T.Y., Xu, J., Qin, T., Xiong, W., Li, H.: LETOR: Benchmark Dataset for Research on Learning to Rank for Information retrieval. In: SIGIR 2007 Workshop, pp. 1192–1199. ACM Press, New York (2007)
11. Jarvelin, K., Kekalainen, J.: IR evaluation methods for retrieving highly relevant documents. In: 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 41–48. ACM Press, New York (2000)
12. Ding, Y.X., Zhou, D., Xiao, M., Dong, L.: Learning to Rank Relational Objects Based on the Listwise Approach. In: 2011 International Joint Conference on Neural Networks, pp. 1818–1824. IEEE Press, New York (2011)

Efficient Semantic Kernel-Based Text Classification Using Matching Pursuit KFDA

Qing Zhang[1], Jianwu Li[2,*], and Zhiping Zhang[3]

[1,3] Institute of Scientific and Technical Information of China,
Beijing 100038, China
[2] Beijing Laboratory of Intelligent Information Technology,
School of Computer Science and Technology,
Beijing Institute of Technology,
Beijing 100081, China
zqbit@tom.com, ljw@bit.edu.cn, zhangzp@istic.ac.cn

Abstract. A number of powerful kernel-based learning machines, such as support vector machines (SVMs), kernel Fisher discriminant analysis (KFDA), have been proposed with competitive performance. However, directly applying existing attractive kernel approaches to text classification (TC) task will suffer semantic related information deficiency and incur huge computation costs hindering their practical use in numerous large scale and real-time applications with fast testing requirement. To tackle this problem, this paper proposes a novel semantic kernel-based framework for efficient TC which offers a sparse representation of the final optimal prediction function while preserving the semantic related information in kernel approximate subspace. Experiments on 20-Newsgroup dataset demonstrate the proposed method compared with SVM and KNN (K-nearest neighbor) can significantly reduce the computation costs in predicating phase while maintaining considerable classification accuracy.

Keywords: Kernel Method, Efficient Text Classification, Matching Pursuit KFDA, Semantic Kernel.

1 Introduction

Text classification (TC) is a challenging problem [1], which aims to automatically assign unlabeled documents to predefined one or more classes according to its contents and is characterized by its inherent high dimensionality and the inevitable existence of polysemy and synonym. To solve those problems, in the last decade, the related studies in document representation, dimensionality reduction and model construction have gained numerous attentions and fruitions [1]. Specifically, this paper mainly focuses on kernel based TC problem.

In recent 20 years, a number of powerful kernel-based learning machines [2], such as support vector machines (SVMs), kernel Fisher discriminant analysis (KFDA), have been proposed and achieved competitive performance in a wide variety of

* Corresponding author.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 382–390, 2011.
© Springer-Verlag Berlin Heidelberg 2011

learning tasks. However, existing attractive kernel approaches are not designed originally for text categorization and often incur huge costs of computation [3]. Kernel method for text is pioneered by Joachims [4] who applies SVM to text classification successfully. Due to the straightforward use of bag of words (BOW) features [4] [5], the semantic relation between words is not taken into consideration. Subsequently, some attentions have been devoted to constructing kernels with semantic information, [6] [7]. Although these attempts take advantage of the modularity in kernel method for improving the performance of TC in the aspects of document representation model and similarity estimation metric, some TC tasks based kernel method are still not practical for the scalability demands which have been increasingly stressed by the advent of large scale and real time applications [8] [9].

The scalable deficiency is inherent for the kernel based methods because the operations on kernel matrix and final optimal solutions largely depend on the whole training examples. To overcome the former problem, previous attempts focus on low rank matrices approximation to make learning algorithms possible to manipulate large scale kernel matrix [10] [11]. For solving the latter one, some approaches straightforwardly deal with the final solution in kernel induced space, such as Burges *et al*. [12] with Reduced-Set method for SVM and Zhang *et al*. [13] using pre-image reconstruction for KFDA while another method adds a constraint to the learning algorithm, which explicitly controls the sparseness of the classifier e.g. Wu *et al*. [14]. Different from the methods discussed above, Diethe *et al*. [15] in 2009, propose a novel sparse KFDA called matching pursuit kernel Fisher discriminant analysis (MPKFDA), which can provide an explicit kernel induced subspace mapping, taking the classification labels into account.

In this paper, taking advantage of the inherent modularity in kernel-based method and the availability of the explicit kernel subspace approximation in Diethe *et al*. 2009 [15], we propose a novel semantic kernel-based framework for efficient TC. In our proposed framework, three different mappings with particular purposes are involved: a) VSM construction mapping, b) semantic kernel space mapping, c) approximate semantic kernel subspace mapping. Using these mappings, the original high dimensional textual data can be transformed into a very low dimensional subspace while maintaining sufficient semantic information and then sparse kernel-based learning model is constructed for efficient testing.

The remainder of this paper is organized as follows. Section 2 introduces kernel method briefly and the proposed method is presented in section 3 followed by the experiments in section 4. The last section concludes this paper.

2 Brief Review of Kernel Methods

Kernel Methods serve as a state-of-the-art framework for all kinds of learning problems which have been successfully introduced into text classification field pioneered by [4]. The main idea behind this approach is the kernel trick using a kernel function to map the data from the original input space into a kernel-induced space implicitly. Then, standard algorithms in input space are performed to solve the kernel induced learning problem reformulated into dot product form substituted by Mercer kernels [2].

The general framework of kernel approach [2] is featured with the modularity, which enable different pattern analysis algorithms to obtain the solution with enhanced ability such as KPCA, which is the kernel version of PCA approach in particular kernel-induced space via diverse kernel functions implicitly.

Given a training set $\{x_1, x_2, \cdots, x_L\}$, a mapping ϕ and a kernel function $k(x_i, x_j)$, all similarity information between input patterns in kernel feature space is entirety preserved in kernel matrix (also called Gram matrix),

$$\mathbf{K} = \left(k(x_i, x_j)\right)_{i,j=1,L} = \left(\left\langle \phi(x_i), \phi(x_j) \right\rangle\right)_{i,j=1,L}. \tag{1}$$

Usually, kernel-based algorithms can seek a linear function solution in feature space [2], as follows

$$f(x) = w'\phi(x) = \sum_{i=1}^{L} \left\langle \alpha_i \phi(x_i), \phi(x) \right\rangle = \sum_{i=1}^{L} \alpha_i k(x_i, x). \tag{2}$$

3 A Novel Semantic Kernel-Based Framework for Efficient TC

As discussed above, the main drawback of this kernel-based TC method is usually lack of sparsity, which is linear proportional to all training samples. It will seriously undermine the classification efficiency on large scale text corpus in predicting phase, especially in real time applications [8] [9].

Framework 1. Semantic Kernel-based Subspace Text Classification.

$d_i \to \phi(d_i) \to R^n \to \phi'(R^n) \to R^k \to \phi''(R^k) \to R^m$

Input: Training text corpus
1: Preprocessing on text corpus
2: Vector space mapping $d_i \to \phi(d_i) \to R^n$
3: Semantic space mapping $R^n \to \phi'(R^n) \to R^k$
4: Low-dim semantic kernel-based subspace approximation mapping $R^k \to \phi''(R^k) \to R^m$
5: Learning model in R^m using any standard classifier
6: Using Step1-4 mapping the test data into low dimensional semantic kernel-based subspace
7: Classifying the mapped data
Output: Result labels for test corpus

To solve this problem, we propose a novel kernel-based framework for TC in this paper, shown in Framework 1. This method extends the general kernel-based framework for text processing. In the following, three mappings for constructing efficient semantic preserved sparse TC model are detailed.

3.1 VSM Construction Mapping

Typical kernel-based algorithms (e.g., SVM) are originally designed for the numerical value vector-based examples in input space. Therefore, vector space model (VSM) [5]

representation for textual data is of key importance in which each document d_i in corpus can be represented as a bag of words (BOW) using the irreversible mapping to *N* dimensional vector space,

$$\phi : d_i \mapsto \phi(d_i) = (tf(t_1, d_i), tf(t_2, d_i), \cdots, tf(t_N, d_i))' \in R^N , \tag{3}$$

where $tf(t_i, d_i)$ is the frequency of the term t_i in d_i and *N* is the terms extracted from the corpus. As a result, we can construct the term-document matrix shown in (4) derived from the corpus containing *L* documents,

$$\mathbf{D}_{VSM} = \begin{pmatrix} tf(t_1, d_1) & \cdots & tf(t_1, d_L) \\ \vdots & \ddots & \vdots \\ tf(t_N, d_1) & \cdots & tf(t_N, d_L) \end{pmatrix}. \tag{4}$$

3.2 Semantic Kernel Space Mapping

Furthermore, using this mapping ϕ, the vector space model-based kernel space can be constructed. The corresponding kernel is $\mathbf{K} = \mathbf{D}'\mathbf{D}$, with the entry $k_{i,j}$ in $\mathbf{K}$ as the inner product between $\phi(d_i)$ and $\phi(d_j)$. More generally, some mappings $\phi' : d \mapsto \phi'(d)$ as the linear transformations of the document can be defined [3] by matrix $\mathbf{p}$,

$$\phi'(d) = \mathbf{P}\phi(d). \tag{5}$$

Subsequently, the kernel matrix becomes

$$\mathbf{K} = \left(\phi(d_i)'\mathbf{P}'\mathbf{P}\phi(d_j)\right)_{i,j=1,L} = \mathbf{D}'\mathbf{P}'\mathbf{P}\mathbf{D}. \tag{6}$$

In addition, Mercer's conditions for $\mathbf{K}$ require that $\mathbf{P}'\mathbf{P}$ should be positive semi-definite.

Under this framework of kernel approach for textual data processing, different choices of P can trigger diverse variants of kernel space. In the case of $\mathbf{P} = \mathbf{I}$ ($\mathbf{I}$ is unit matrix), vector space model (VSM) induced kernel space is established, which maps each document to a vector representation as in (3). However, the main limitation of such approach lies in the absence of semantic information, which is incapable of addressing the problem of synonymy and polysemy [3].

In order to solve ambiguity in similarity measure, various methods have been developed for the extraction of semantic information in large scale corpus through textual contents such as Latent Semantic Indexing (LSI) [16], or exterior resources such as semantic networks in a hierarchical structure [17] [18]. All these methods can be incorporated into our framework.

In this paper, we employ the LSI method to construct semantic kernel as described in Cristianini *etc al.* [3] for our proposed framework to overcome the semantic deficiency problem. LSI is a transformed-based feature reduction approach which offers the possibility of mapping the document in VSM into a semantic subspace defined by several concepts using Singular Value Decomposition (SVD) in an unsupervised way [16]. In that low-dimensional concept-based subspace, the similarity between documents can reflect the semantic structures by taking words co-occurrence information into account. More precisely, the term document matrix derived from (4) is decomposed using SVD,

$$\mathbf{D} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{'}, \tag{7}$$

where the columns of matrix $\mathbf{U}$ and $\mathbf{V}$ are the eigenvectors of $\mathbf{DD}^{'}$ and $\mathbf{D}^{'}\mathbf{D}$ respectively, $\boldsymbol{\Sigma}$ is a diagonal matrix with nonnegative real diagonal singular values sorted in decreasing order.

The key to building LSI kernel is to find the matrix $\mathbf{P}$ defined by the mapping $\phi^{'}: d \mapsto \phi^{'}(d)$. For LSI case, the concept subspace is spanned by the first *k* columns of $\mathbf{U}$, which form the matrix $\mathbf{P}$,

$$\mathbf{P} = \mathbf{U_k}' = (u_1 u_{2,\cdots,} u_k)'. \tag{8}$$

Hence the LSI kernel mapping is $\phi^{'}: d \mapsto \phi^{'}(d) = \mathbf{U_k}'\phi(d)$ and the kernel matrix is

$$\mathbf{K} = \left(\phi(d_i)'\mathbf{U_k}\mathbf{U_k}'\phi(d_j)\right)_{i,j=1,L} = \mathbf{D}'\mathbf{U_k}\mathbf{U_k}'\mathbf{D}. \tag{9}$$

3.3 Approximate Semantic Kernel Subspace Mapping

The third mapping is crucial for our final sparse model construction. However, previous efforts addressing kernel-induced subspace approximation mainly focus on training phase using low-rank matrix approximation [10] [11] with the purpose to simplify specific optimizing process, which can not contribute to our third mapping. Although the approaches e.g. [12] [13] deal with our problem directly, those need a full final model in advance.

Recently, Matching Pursuit Kernel Fisher Discriminant Analysis proposed by [15] in 2009 offers us a new approach to finding a low dimensional space by kernel subspace approximation, the fundamental principle of which is the use of Nyström method of low rank approximation for the Gram matrix in a greedy fashion. MPKFDA is suitable to our problem because it can find the explicit kernel-based subspace in which any standard machine learning can be applied. Thus, we incorporate MPKFDA into our framework such that the data in semantic kernel space can be projected into its approximation subspace with low dimensionality.

We assume $\mathbf{X}$ is the data matrix containing the projected data in a semantic kernel induced space, which are stored as row vectors and $\mathbf{K}[i, j] = \langle x_i, x_j \rangle$ are

the entries of kernel matrix $\mathbf{K}$. The notation of $\mathbf{K}[:,i]$ and $\mathbf{K}[\mathbf{i},\mathbf{i}]$ represent the i^{th} column of $\mathbf{K}$ and the square matrix defined by a set of indices $\mathbf{i}=\{i_1,\ldots,i_m\}$, respectively.

According to [15], the final subspace projection is through $\mathbf{K}[:,\mathbf{i}]\mathbf{R}'$ as a new data matrix in the low dimensional semantic kernel induced subspace, which is derived via applying the Nyström method of low rank approximation for the Kernel matrix,

$$\begin{aligned}\tilde{\mathbf{K}} &= \mathbf{K}[:,\mathbf{i}]\mathbf{K}[\mathbf{i},\mathbf{i}]^{-1}\mathbf{K}[:,\mathbf{i}]' \\ &= \mathbf{K}[:,\mathbf{i}]\mathbf{R}'\mathbf{R}\mathbf{K}[:,\mathbf{i}]',\end{aligned} \tag{10}$$

where $\mathbf{R}$ is the Cholesky decomposition of $\mathbf{K}[\mathbf{i},\mathbf{i}]^{-1}=\mathbf{R}'\mathbf{R}$.

Moreover, this kernel matrix approximation can be viewed as a form of covariance matrix in this space,

$$\tilde{\sum}_k = \mathbf{R}\mathbf{K}[:,\mathbf{i}]'\mathbf{K}[:,\mathbf{i}]\mathbf{R}'. \tag{11}$$

In order to seek a set $\mathbf{i}=\{i_1,\ldots,i_k\}$, the iterative greedy procedure is performed to seek i_k in the k^{th} round by choosing the i_k which leads to maximization of the Fisher discriminant ratio,

$$\max_w J(w)=\frac{(\mu_w^+-\mu_w^-)^2}{(\sigma_w^+)^2+(\sigma_w^-)^2+\lambda\|w\|^2}, \tag{12}$$

where μ_w^+ and μ_w^- are the means of the projection of the positive and negative examples respectively onto the direction w and σ_w^+, σ_w^- are the corresponding standard deviations and then deflate the $\mathbf{K}$,

$$\mathbf{K}=\left(\mathbf{I}-\frac{\mathbf{K}[:,i_k]\mathbf{K}[:,i_k]'}{\mathbf{K}[:,i_k]'\mathbf{K}[:,i_k]}\right)\mathbf{K}, \tag{13}$$

ensuring the remaining potential basis vectors are orthogonal to those bases already picked.

The maximization rule is

$$\max_i \rho_i=\frac{\mathbf{e}_i'\mathbf{X}\mathbf{X}'\mathbf{y}\mathbf{y}\mathbf{X}\mathbf{X}'\mathbf{e}_i}{\mathbf{e}_i'\mathbf{X}\mathbf{X}'\mathbf{B}\mathbf{X}\mathbf{X}'\mathbf{e}_i}=\frac{\mathbf{K}[:,i]'\mathbf{y}\mathbf{y}'\mathbf{K}[:,i]}{\mathbf{K}[:,i]'\mathbf{B}\mathbf{K}[:,i]}, \tag{14}$$

which is derived via substituting $\mathbf{w}=\mathbf{X}'\mathbf{e}_i$ in the following equation as the FDA problem [2]

$$\mathbf{w}=\max_{\mathbf{w}}\frac{\mathbf{w}'\mathbf{X}'\mathbf{y}\mathbf{y}'\mathbf{X}\mathbf{w}}{\mathbf{w}'\mathbf{X}'\mathbf{B}\mathbf{X}\mathbf{w}}, \tag{15}$$

where $\mathbf{e}_i$ is the i^{th} unit vector, $\mathbf{y}$ is the label vector with +1 or -1, and $\mathbf{B}=\mathbf{D}-\mathbf{C}^+-\mathbf{C}^-$ as defined in [2].

After finding the low dimensional semantic kernel induced subspace, all the training data are projected into this space using $\mathbf{K}[:,\mathbf{i}]\mathbf{R}'$ recomputed by the samples indexed in the optimal set $\mathbf{i} = \{i_1, \ldots, i_k\}$ as our third mapping. Then, we can acquire the final classification model for testing phase by solving the linear FDA problem within this space. See [15] for details.

4 Experiments

4.1 Experimental Settings

In our experiments, 20-Newsgroups (20NG) dataset [19] is used to evaluate our proposed method compared with the SVM with linear kernel and KNN in LSI feature space. To make the task more challenging, we select the most similar sub-topics in the lowest level in 20NG as our six binary classification problems listed in Table 1 with the approximate 5 fold cross validation scheme. After some preprocessing procedures including stop words filtering and stemming, BOW model is created (4). The average dimensionalities of BOW generated are also shown in Table 1. It is noted that KNN is implemented in the nearest neighbor way and the LSI space holds 100 dimensions.

Table 1. Six Binary Classification Problem Settings on 20-Newsgroups Dataset

ID	Class-P	Class-N	N-Train	N-Test	D-BOW
S-1	talk.politics.guns	talk.politics.mideast	1110	740	12825
S-2	talk.politics.guns	talk.politics.misc	1011	674	10825
S-3	talk.politics.mideast	talk.politics.misc	1029	686	12539
S-4	rec.autos	rec.motorcycles	1192	794	9573
S-5	com.sys.ibm.pc.hardware	com.sys.mac.hardware	1168	777	8793
S-6	sci.electronics	sci.space	1184	787	10797

4.2 Experimental Results and Discussions

The experimental (best average) results are shown in Table 2 for the proposed method (SKF-ETC), LSI Kernel-SVM and KNN. Table 2 demonstrates our method can significantly decrease the number of the bases in the final solution. Specially, we can find KNN needs all the training samples to predict unknown patterns, and although SVM can decrease the number of training data responsible for constructing the final model by using support vectors (SV), the total number of SV is still large for large scale TC tasks. On the contrary, SKF-ETC can only hold very small number of bases spanning the approximate semantic kernel-based subspace for text classification. Moreover, as shown in Fig.1 to Fig.6, those experimental findings as well as the inherent convergence property of MPKFDA [15] to full solution can guarantee the effectiveness of the proposed SKF-ETC.

Table 2. Results on Six Binary Classifications for SKF-ETC, SVM and KNN

Task ID	SKF-ETC		LSI Kernel-SVM		LSI-KNN	
	N-Basis	Accuracy	N-SV	Accuracy	N-Train	Accuracy
S-1	28	0.9108	107	0.9572	1110	0.9481
S-2	17	0.8026	231	0.8420	1011	0.8234
S-3	19	0.8772	128	0.9189	1029	0.9131
S-4	25	0.8836	192	0.9153	1192	0.8239
S-5	31	0.7863	392	0.8069	1168	0.7127
S-6	28	0.8996	123	0.9432	1184	0.8694

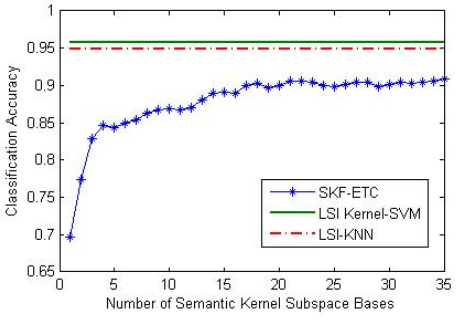


Fig. 1. ID-S-1

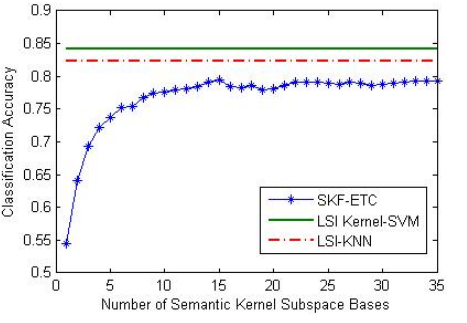


Fig. 2. ID-S-2

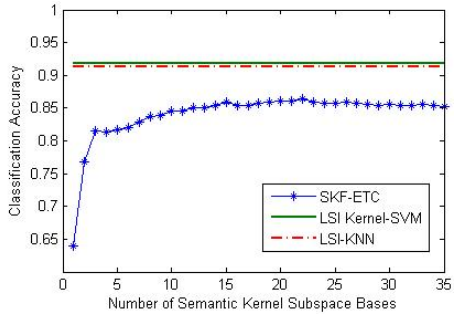


Fig. 3. ID-S-3

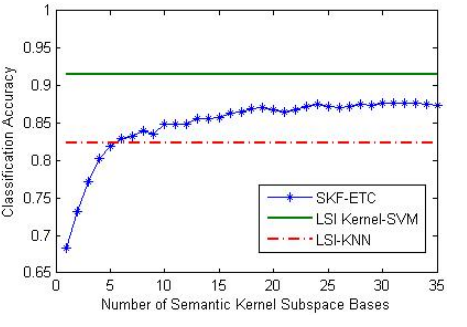


Fig. 4. ID-S-4

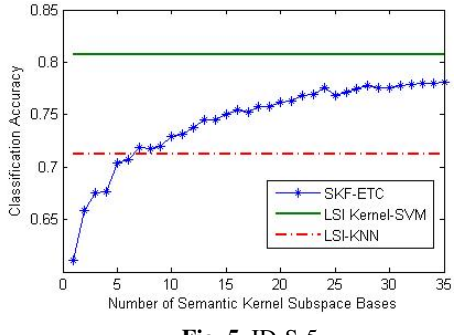


Fig. 5. ID-S-5

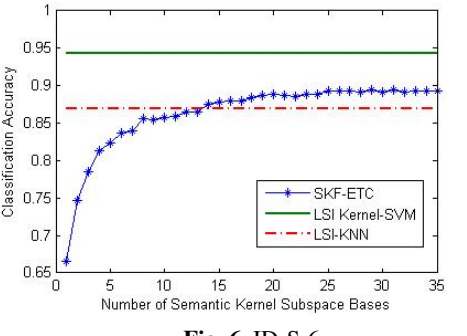


Fig. 6. ID-S-6

5 Conclusions

The urgent requirements [8] [9] for speeding up the prediction for TC are demanded by numerous large scale and real-time applications using kernel-based approaches. In order to solve this problem, this paper proposes a novel framework for semantic kernel-based efficient TC. In fact, any other semantic kernels beyond LSI can be incorporated into our framework for TC with modularity, which also characterizes our proposed method at the scalability aspect.

References

1. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Comput. Surv. (CSUR) 34(1), 1–47 (2002)
2. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York (2004)
3. Cristianini, N., Shawe-Taylor, J., Lodhi, H.: Latent Semantic Kernels. J. Intell. Inf. Syst. (JIIS) 18(2-3), 127–152 (2002)
4. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
5. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Commun. ACM (CACM) 18(11), 613–620 (1975)
6. Kandola, J., Shawe-Taylor, J., Cristianini, N.: Learning Semantic Similarity. In: NIPS, pp. 657–664 (2002)
7. Tsatsaronis, G., Varlamis, I., Vazirgiannis, M.: Text Relatedness Based on a Word Thesaurus. J. Artif. Intell. Res (JAIR) 37, 1–39 (2010)
8. Wang, H., Chen, Y., Dai, Y.: A Soft Real-Time Web News Classification System with Double Control Loops. In: Fan, W., Wu, Z., Yang, J. (eds.) WAIM 2005. LNCS, vol. 3739, pp. 81–90. Springer, Heidelberg (2005)
9. Miltsakaki, E., Troutt, A.: Real-time Web Text Classification and Analysis of Reading Difficulty. In: The Third Workshop on Innovative Use of NLP for Building Educational Applications at ACL, pp. 89–97 (2008)
10. Smola, A.J., Schökopf, B.: Sparse Greedy Matrix Approximation for Machine Learning. In: ICML, pp. 911–918 (2000)
11. Fine, S., Scheinberg, K.: Efficient SVM Training Using Low-Rank Kernel Representations. Journal of Machine Learning Research (JMLR) 2, 243–264 (2001)
12. Burges, C.J.C.: Simplified Support Vector Decision Rules. In: ICML, pp. 71–77 (1996)
13. Zhang, Q., Li, J.: Constructing Sparse KFDA Using Pre-image Reconstruction. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) ICONIP 2010, Part II. LNCS, vol. 6444, pp. 658–667. Springer, Heidelberg (2010)
14. Wu, M., Schölkopf, B., Bakir, G.: Building Sparse Large Margin Classifiers. In: ICML, pp. 996–1003 (2005)
15. Diethe, T., Hussain, Z., Hardoon, D.R., Shawe-Taylor, J.: Matching Pursuit Kernel Fisher Discriminant Analysis. In: AISTATS, pp. 121–128 (2009)
16. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by Latent Semantic Analysis. JASIS 41(6), 391–407 (1990)
17. Wang, P., Domeniconi, C.: Building Semantic Kernels for Text Classification Using Wikipedia. In: KDD, pp. 713–21 (2008)
18. Hu, X., Zhang, X., Lu, C., Park, E.K., Zhou, X.: Exploiting Wikipedia as External Knowledge for Document Clustering. In: KDD, pp. 389–396 (2009)
19. 20 Newsgroups Dataset, `http://people.csail.mit.edu/jrennie/20Newsgroups/`

Introducing a Novel Data Management Approach for Distributed Large Scale Data Processing in Future Computer Clouds

Amir H. Basirat and Asad I. Khan

Clayton School of IT, Monash University
Melbourne, Australia
{Amir.Basirat,Asad.Khan}@monash.edu

Abstract. Deployment of pattern recognition applications for large-scale data sets is an open issue that needs to be addressed. In this paper, an attempt is made to explore new methods of partitioning and distributing data, that is, resource virtualization in the cloud by fundamentally re-thinking the way in which future data management models will need to be developed on the Internet. The work presented here will incorporate content-addressable memory into Cloud data processing to entail a large number of loosely-coupled parallel operations resulting in vastly improved performance. Using a lightweight associative memory algorithm known as Distributed Hierarchical Graph Neuron (DHGN), data retrieval/processing can be modeled as pattern recognition/matching problem, conducted across multiple records and data segments within a single-cycle, utilizing a parallel approach. The proposed model envisions a distributed data management scheme for large-scale data processing and database updating that is capable of providing scalable real-time recognition and processing with high accuracy while being able to maintain low computational cost in its function.

Keywords: Distributed Data Processing, Neural Network, Data Mining, Associative Computing, Cloud Computing.

1 Introduction

With the advent of distributed computing, distributed data storage and processing capabilities have also contributed to the development of cloud computing as a new paradigm. Cloud computing can be viewed as a pay-per-use paradigm for providing services over the Internet in a scalable manner. The cloud paradigm takes on two different data management perspectives, namely storage and applications. With different kinds of cloud-based applications and a variety of database schemes, it is critical to consider integration between these two entities for seamless data access on cloud. Nevertheless, this integration has yet to be fully-realized.

Existing frameworks such as MapReduce [1] and Hadoop [2] involve isolating basic operations within an application for data distribution and partitioning. This limits their applicability to many applications with complex data dependency considerations. According to Shiers [3], “it is hard to understand how data intensive

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 391–398, 2011.
© Springer-Verlag Berlin Heidelberg 2011

applications, such as those that exploit today's production grid infrastructures, could achieve adequate performance through the very high-level interfaces that are exposed in clouds". In addition to this complexity, there are other underlying issues that need to be addressed properly by any data management scheme deployed for clouds. Some of these concerns are highlighted by Abadi [4] including: capability to parallelize data workload, security concerns as a result of storing data at an untrusted host, and data replication functionality. The new surge in interest for cloud computing is accompanied with the exponential growth of data sizes generated by digital media (images/audio/video), web authoring, scientific instruments, and physical simulations. Thus the question, how to effectively process these immense data sets is becoming increasingly important. Also, the opportunities for parallelization and distribution of data in clouds make storage and retrieval processes very complex, especially in facing with real-time data processing [5]. With these particular aspects in mind, we would like to investigate novel schemes that can efficiently partition and distribute complex data for large-scale data processing in clouds. For this matter, loosely coupled associative techniques, not considered so far, hold the key to efficient partitioning and distributing such data in the clouds and its fast retrieval.

2 Distributed Data Management

The efficiency of the cloud system in dealing with data intensive applications through parallel processing, essentially lies in how data is partitioned among nodes, and how collaboration among nodes are handled to accomplish a specific task. Our proposal is based on a special type of Associative Memory (AM) model, which is readily implemented within distributed architectures. AM is a subset of artificial neural networks, which utilizes the benefits of content-addressable memory (CAM) [6] in microcomputers. AM is also one of the important concepts in associative computing. In this regard, the development of associative memory (AM) has been largely influenced by the evolution of neural networks. Some of the established neural networks that have been used in pattern recognition applications include: Hopfield's Associative Memory network [7], bidirectional associative memory (BAM) [8], and fuzzy associative memory (FAM) [9]. These associative memories generally apply the Hebbian learning rule or kernel-based learning approach. Thus, these AMs remain susceptible to the well-known limits of these learning approaches in terms of scalability, accuracy, and computational complexity. It has been suggested in the literature that graph-based algorithms provide various tools for graph-matching pattern recognition [10], while introducing universal representation formalism [11]. The main issue with these approaches lies in the significant increase in the computational expenses of the deployed methods as a result of increase in the size of the pattern database [12]. This increase puts a heavy practical burden on deployment of those algorithms in clouds for data-intensive applications, and real-time data processing and database updating. Hierarchical structures in associative memory models are of interest as these have been shown to improve the rate of recall in pattern recognition applications [13]. As we know, existing data access mechanisms for cloud computing such as MapReduce has proven the ability for parallel access approach to be performed on cloud infrastructure [14]. Thus, our aim is to apply a

data access scheme that enables data retrieval to be conducted across multiple records and data segments within a single-cycle, utilizing a parallel approach. Using a lightweight associative memory algorithm known as Distributed Hierarchical Graph Neuron (DHGN), data retrieval/processing can be modeled as a pattern recognition/matching problem, and tackled in a very effective and efficient manner. DHGN extends the functionalities and capabilities of Graph Neuron (GN) and Hierarchical Graph Neuron (HGN) algorithms.

2.1 Graph Neuron (GN) for Scalable Pattern Recognition

GN pattern representation simply follows the representation of patterns in other graph-matching based algorithms. Each GN in the network holds a (value, position) pair information of elements that constitutes the pattern. In correspondence towards graph-based structure, each GN acts as a vertex that holds pattern element information (in the form of value or identification) while the adjacency communication between two or more GNs is represented by the edge of a graph. Message communications in GN network are restricted only to the adjacent nodes (of the array), hence there is no increase in the communication overheads with corresponding increases in the number of nodes in the network [15]. GN recognition process involves the memorization of adjacency information obtained from the edges of the graph (See Figure 1).

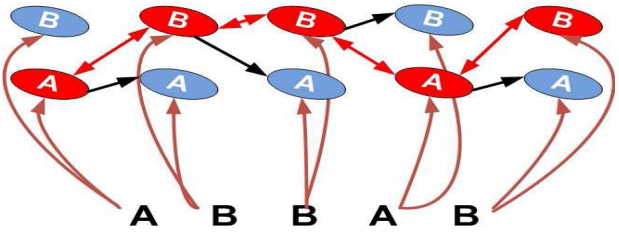


Fig. 1. GN network activation from input pattern "ABBAB"

2.2 Crosstalk Issue in Graph Neuron

GN's limited perspective on overall pattern information would result in a significant inaccuracy in its recognition scheme. As the size of the pattern increases, it is more difficult for a GN network to obtain an overview of the pattern's composition. This produces incomplete results, where different patterns having similar sub-pattern structure lead to false recall. Let us suppose that there is a GN network which can allocate 6 possible element values, e.g. u, v, w, x, y, and z, for a 5-element pattern. A pattern uvwxz, followed by zvwxy is introduced. These two patterns would be stored by the GN array. Next, we introduce the pattern uvwxy, this will produce a recall. Clearly the recall is false since the last pattern does not match the previously stored patterns. The reason for this false recall is that a GN node only knows of its own value and its adjacent GN values. Hence, the input patterns in this case will be stored as segments uv, uvw, vwx, wxy, xy. The latest input pattern, though different from the two previous patterns, contain all the segments of the previously stored patterns

In order to solve the issue of the crosstalk due to the limited perspective of GNs, the capabilities of perceiving GN neighbors in each GN is expanded in Hierarchical Graph Neuron (HGN) to prevent pattern interference. This is achieved by having higher layers of GN neurons that oversee the entire pattern information. Hence, it will provide a bird's eye view of the overall pattern. Figure 2 shows the hierarchical layout of HGN for binary pattern with size of 7 bits.

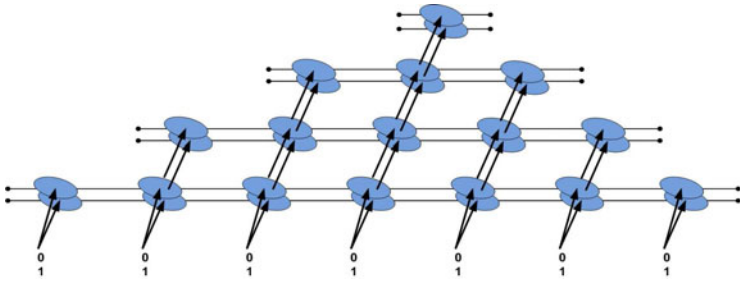


Fig. 2. Hierarchical Graph Neuron (HGN) with binary pattern of size 7 bits

2.3 Hierarchical Graph Neuron (GN) for Scalable Pattern Recognition

Each GN (except the ones on the edges) must be able to monitor the condition of not just the adjacent columns, but also the ones further away. This approach would however cause a communication bottleneck as the size of array increases. The problem is solved by introducing higher levels of GN arrays. These arrays receive inputs from their lower arrays. The array on the base level receives the actual pattern inputs. Higher level arrays are added until a single column is needed to oversee the underlying array. The number of GN at the base level array must, therefore, be an odd number in order to end up with a single column within the top array. In turn, the GN within a higher array only communicates with the adjacent columns at their level. Each higher level GN receives an input from the underlying GN in the lower array. The value sent by the GN at the base level is an index of the unique pair value p(left, right), i.e. the bias entry, of the current pattern. The index starts from unity and is incremented by one. The base level GN sends the index of every recorded or recalled pair value p(left, right) to their corresponding higher level GN. The higher level GN can thus provide a more authoritative assessment of the input pattern.

2.4 Distributed Hierarchical Graph Neuron (DHGN)

HGN can be extended by dividing and distributing the recognition processes over the network. This distributed scheme minimizes the number of processing nodes by reducing the number of levels within the HGN. DHGN is in fact a single-cycle learning associative memory (AM) algorithm for pattern recognition. DHGN employs the collaborative-comparison learning approach in pattern recognition. It lowers the complexity of recognition processes by reducing the number of processing nodes. In addition, as depicted in Figure 3, pattern recognition using DHGN algorithm is improved through a two-level recognition process, which applies recognition at sub-pattern level and then recognition at the overall pattern level.

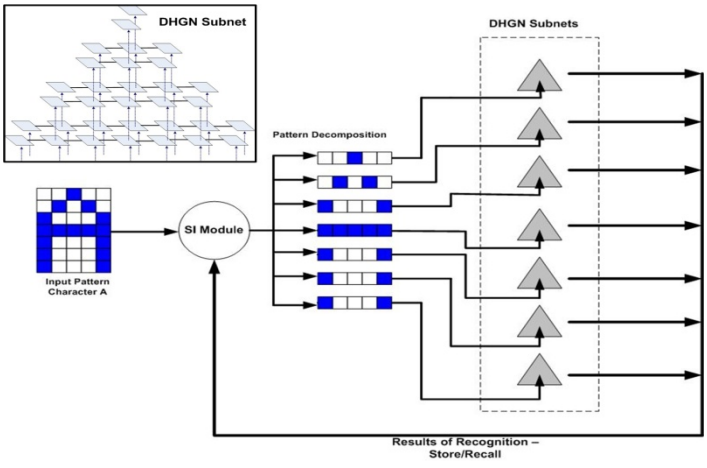


Fig. 3. DHGN distributed pattern recognition architecture

The recognition process performed using DHGN algorithm is unique in a way that each subnet is only responsible for memorizing a portion of the pattern (rather than the entire pattern). A collection of these subnets is able to form a distributed memory structure for the entire pattern. This feature enables recognition to be performed in parallel and independently. The decoupled nature of the sub-domains is the key feature that brings dynamic scalability to data management within cloud computing. Figure 4 shows the divide-and-distribute transformation from a monolithic HGN composition (top) to a DHGN configuration for processing the same 35-bit patterns (bottom).

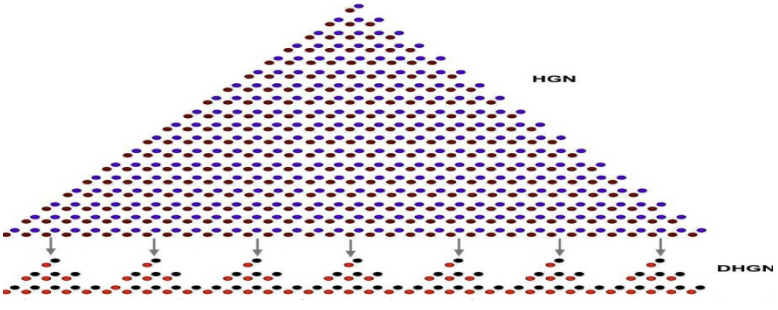


Fig. 4. Transformation of HGN structure (top) into an equivalent DHGN structure (bottom)

The base of the HGN structure in Figure 4 represents the size of the pattern. Note that the base of HGN structure is equivalent to the cumulative base of all the DHGN subnets/clusters. This transformation of HGN into equivalent DHGN composition allows on the average 80% reduction in the number of processing nodes required for the recognition process. Therefore, DHGN is able to substantially reduce the computational resource requirement for pattern recognition process – from 648 processing nodes to 126 for the case shown in Figure 4.

3 Tests and Results

In order to validate the proposed scheme, a cloud computing environment is formulated for executing the proposed algorithm over very large number of GN nodes. The simulation program deals with data records as patterns and employs Distributed Hierarchical Graph Neuron (DHGN) to process those patterns. Since our proposed model relies on communications among adjacent nodes, the decentralized content location scheme is implemented for discovering adjacent nodes in minimum number of hops. A GN-based algorithm for optimally distributing DHGN subnets (clusters or sub-domains) among the cloud nodes is also deployed to automate the boot-strapping of the distributed application over the network. After initial network training, the cloud will be fed with new data records (patterns) and the responsible processing nodes will process the data record to see if there is an exact match or similar match (with distortion) for that record. The input pattern can also be defined with various levels of distortion rate. In fact, DHGN exhibits unique functional performance with regards to handling distorted data records (patterns) as is the norm in many cloud environments.

Figure 5 illustrates parsing times at sub-pattern level. As clearly depicted, with an increase in the length of the sub-pattern, average parsing time also increases, however this increase is not substantial due to the layered and distributed structure of DHGN. This significant effect is at the heart of DHGN scalability, making it remarkably suitable for large-scale data processing in clouds.

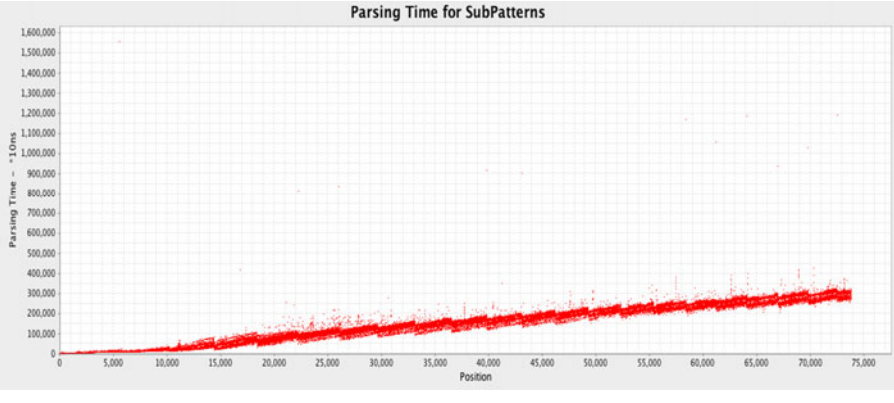


Fig. 5. Average parsing time for sub-patterns as the length of sub-patterns increases

3.1 Superior Scalability

Another important aspect of DHGN is that it can remain highly scalable. In fact, its response time to store or recall operations is not affected by an increase in the size of the stored pattern database. The flat slope in Figure 6 shows that the response times remain insensitive to the increase in stored patterns, representing the high scalability of the scheme. Hence, the issue of computational overhead increase due to the

increase in the size of pattern space or number of stored patterns, as is the case in many graph-based matching algorithms will be alleviated in DHGN, while the solution can be achieved within fixed number of steps of single cycle learning and recall.

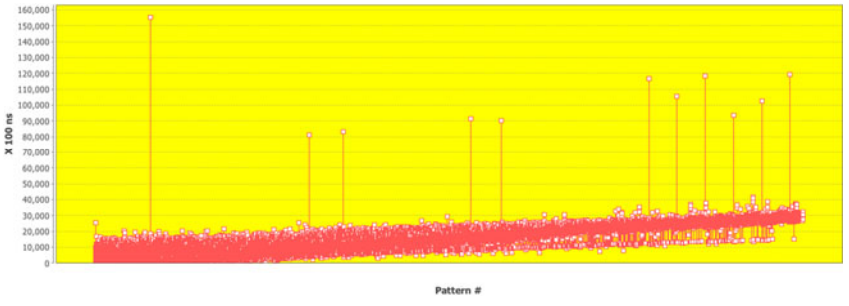


Fig. 6. Response time as more and more patterns are introduced in to the network

3.2 Recall Accuracy

The DHGN data processing scheme continues to improve its accuracy as more and more patterns are stored. It can be seen from Figure 7 that the accuracy of DHGN in recognizing previously stored patterns remains consistent and in some cases shows significant increase as more and more patterns are stored (greater improvement with more one-shot learning experiences). The DHGN data processing achieved above 80% accuracy in our experiments after all the 10,000 patterns (with noise) had been presented.

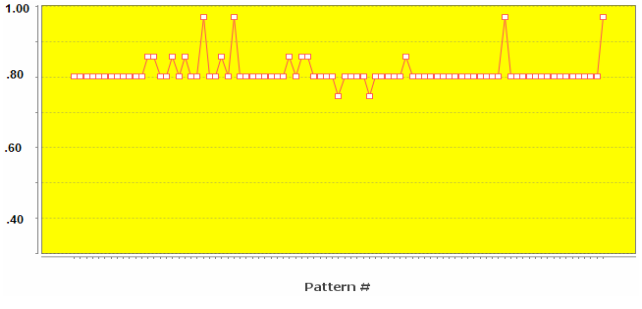


Fig. 7. Recall accuracy for a DHGN composition as more and more patterns are introduced into the network

4 Conclusion

In contrast with hierarchical models proposed in the literature, DHGN's pattern recognition capability and the small response time, that remains insensitive to the increases in the number of stored patterns, makes this approach ideal for Clouds.

Moreover, the DHGN does not require definition of rules or manual interventions by the operator for setting of thresholds to achieve the desired results, nor does it require heuristics entailing iterative operations for memorization and recall of patterns. In addition, this approach allows induction of new patterns in a fixed number of steps. Whilst doing so it exhibits a high level of scalability i.e. the performance and accuracy do not degrade as the number of stored patterns increase over time. Furthermore all computations are completed within the pre-defined number of steps and as such the approach implements one-shot, single-cycle or single-pass, learning.

References

1. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters, In: Proceedings of 6th Conference on Operating Systems Design & Implementation (2004)
2. Hadoop, `http://lucene.apache.org/hadoop`
3. Shiers, J.: Grid today, clouds on the horizon. Computer Physics, 559–563 (2009)
4. Abadi, D.J.: Data Management in the Cloud: Limitations and Opportunities. Bulletin of the Technical Committee on Data Engineering, 3–12 (2009)
5. Szalay, A., Bunn, A., Gray, J., Foster, I., Raicu, I.: The Importance of Data Locality in Distributed Computing Applications, In: Proc. of the NSF Workflow Workshop (2006)
6. Chisvin, L., Duckworth, J.R.: Content-addressable and associative memory: alternatives to the ubiquitous RAM. IEEE Computer 22, 51–64 (1989)
7. Hopfield, J.J., Tank, D.W.: Neural Computation of Decisions in Optimization Problems. Biological Cybernetics 52, 141–152 (1985)
8. Kosko, B.: Bidirectional Associative Memories. IEEE Transactions on Systems and Cybernetics 18, 49–60 (1988)
9. Kosko, B.: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall, NJ (1992)
10. Luo, B., Hancock, E.R.: Structural graph matching using the EM algorithm and singular value decomposition. IEEE Trans. Pattern Anal. Machine Intelligence 23(10), 1120–1136 (2001)
11. Irniger, C., Bunke, H.: Theoretical Analysis and Experimental Comparison of Graph Matching Algorithms for Database Filtering. In: Hancock, E.R., Vento, M. (eds.) GbRPR 2003. LNCS, vol. 2726, pp. 118–129. Springer, Heidelberg (2003)
12. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
13. Ohkuma, K.: A Hierarchical Associative Memory Consisting of Multi-Layer Associative Modules. In: Proc. of 1993 International Joint Conference on Neural Networks (IJCNN 1993), Nagoya, Japan (1993)
14. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large Clusters. Communications of the ACM, 107–113 (2008)
15. Khan, A.I., Mihailescu, P.: Parallel Pattern Recognition Computations within a Wireless Sensor Network. In: Proceedings of the 17th International Conference on Pattern Recognition. IEEE Computer Society, Cambridge (2004)

PatentRank: An Ontology-Based Approach to Patent Search

Ming Li, Hai-Tao Zheng*, Yong Jiang, and Shu-Tao Xia

Tsinghua-Southampton Web Science Laboratory, Graduate School at Shenzhen, Tsinghua University, China
liming.sd.china@gmail.com,
{zheng.haitao,jiangy,xiast}@sz.tsinghua.edu.cn

Abstract. There has been much research proposed to use ontology for improving the effectiveness of search. However, there are few studies focusing on the patent area. Since patents are domain-specific, traditional search methods may not achieve a high performance without knowledge bases. To address this issue, we propose PatentRank, an ontology-based method for patent search. We utilize International Patent Classification (IPC) as an ontology to enable computer to better understand the domain-specific knowledge. In this way, the proposed method is able to well disambiguate user's search intents. And also this method discovers the relationship between patents and employs it to improve the ranking algorithm. The empirical experiments have been conducted to demonstrate the effectiveness of our method.

Keywords: Semantic Search, Lucene, Patent Search, Ontology, IPC.

1 Introduction

Due to the great advancement of Internet, Information Explosion has become a severe issue today. People may find it difficult to locate what they really want among mass-data in the Web, which drives a number of scholars to commit themselves to the studying of information searching techniques, a lot of approaches have been proposed, and some search engines have been developed and commercialized consequently, such as the most outstanding Google. However, many questions remain left with no answer, even in terms of the tremendous searching power of Google.

With the emerging of "Semantic Web" theory and technology, research on the search methods under the Semantic Web architecture is quite applicable and promising owing to the attributes of Semantic Web, e.g. the ability to improve the precision by means of getting the machine to understand user's search intent and the specific meaning in the context of query space. In this study, we present a Semantic Search system in the patent area. The attributes of patent area are taken into consideration in this choice, namely with expanding of patent database size, it is a

* Corresponding author.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 399–405, 2011.
© Springer-Verlag Berlin Heidelberg 2011

tough problem to an applicant, especially to a non-expert user, who wants to confirm whether his/her invention has been registered by searching the patent database, while the lack of comprehensive patent search algorithm and professional patent search engine aggravate this morass. Thus we develop a novel approach to patent search in our system and extensive experiments are conducted to verify its effectiveness.

The rest of this paper is organized as follows. In next Section we take a brief overview over existing work on Semantic Search and its sorting algorithm. We discuss the detailed methodology to build our system in Section 3. Section 4 describes evaluation. Finally Section 5 presents our conclusions and future work.

2 Related Work

To our best knowledge, the concept of Semantic Search was firstly put forth in [1], which distinguished between two different kinds of searches-Navigational Searches and Research Search. With the advent of the Semantic Web, research in this area is flourishing. Many scholars have made great progress in various branches of Semantic Web, among which Semantic Search is a significant one [2][3]. Current web search engines do not work well with Semantic Web documents, for they are designed to address the traditional unstructured text. Thus research of search on semi-structured or structured documents has emerged tremendously in recent years [4][5][6][7]. [4] presented an entity retrieval system-SIREn based on a node indexing scheme, this system was able to index and query very large semi-structured datasets. [5] proposed an approach to XML keyword search and presents a XML TF*IDF ranking strategy to support their work. Ranking is a key part in a search system, thus ranking algorithm in Semantic Web is one of the fundamental research points[8][9]. [10] presented a technique for ranking based on the Semantic Web resource importance. Many scholars contribute to the research on retrieval of domain-specific documents [11][12].

3 Methodology

3.1 Hypothesis

In order to maximize users' fulfillment of their search intents, several principles must be taken into account in searching process, which are disambiguation of query expression, accuracy and comprehensiveness of search results. With the aim of achieving the above principles we develop our approach based on three guidelines: **Guideline 1**: The ambiguity of keyword in the query expression could be reduced greatly if it is confined to a certain area (or a specific domain); **Guideline 2**: Words or phrases that match the query expression should contribute differently to the ranking according to the position (field) they appear in the patent document. **Guideline 3**: The patent which has same keyphrases and IPCs with those ranks higher in the search results should be elevated in ranking.

3.2 Ontology-Based Ranking

In our system, we use Lucene [13] as the search engine and baseline, at the core of Lucene's logical architecture is the idea of a document containing fields of text, this feature allows Lucene's API to be independent of the file format; and the term "field" has been mentioned in Guideline 2, which could bring you much flexibility in precisely control how Lucene will index the field's value and convenience if you want to boost a certain part of a document.

Documents that matched the query are ranked according to the following scoring formula [14]:

$$\text{score}(q,d) = \text{coord}(q,d) \cdot \text{queryNorm}(q) \cdot \sum_{t \text{ in } q} (\text{tf}(t \text{ in } d) \cdot \text{idf}(t)^2 \cdot t.\text{getBoost}() \cdot \text{norm}(t,d))$$

Where:

t: term(search keyword), **q:** query, **d:** document

coord(q,d): is a score factor based on how many of the query terms are found in the specified document.

queryNorm(q): is a normalizing factor used to make scores between queries comparable in search time, it does not affect document ranking.

t.getBoost(): is a search time boost of term t in the query q as specified in the query expression, or as set by application calls to a method.

norm(t,d): encapsulates a few (indexing time) boost and length factors.

IPC is the semantic annotation data (ontology) of patents, in our system, we index IPC documents as well as patent documents respectively, and in searching process, we use the same query expression to search both the patents and the IPCs, thereafter score them separately, at last we sort the results by means of combining the patent score and its IPC score. Note that some patents have several IPCs (as result they have several IPC scores), it means that those patents could be categorized into more than one category, in this occasion we combine its highest IPC score with the patent's. Use an equation to express this:

$$\text{Score}(p) = (1-\alpha)\text{score}(q, d_{\text{patent}}) + \alpha\text{Max}(\text{score}(q, d_{\text{IPC in patent}})) \quad (1)$$

where

p: denotes the patent

***α*:** is an adjusting parameter, its range is [0, 1).

3.3 Reranking Based on Similarity

In our system we first apply Maui [14] to extract the keyphrases from patents. Maui is a general algorithm for automatic topical indexing; it builds on the Kea system [15] that is employing the Naive Bayes machine learning method. Maui enhances Kea's

successful machine learning framework with semantic knowledge retrieved from Wikipedia, new features, and a new classification model.

Next, we propose a novel ranking method, which is in fact a reranking process based on the initial search results due to the patents' scores drawn from Formula (1). Patents with same IPC and keyphrases are assumed to be quite similar with each other and could be classified into one group, if one of the group members interests user, the others may also do.

In practice we choose a certain number of patents (which might satisfy the user with high possibility, e.g. the first ten) from the initial results as roots, and then we define each root as a single source to build a directed acyclic patent graph respectively with the other patents in the search results. In the patent graph, nodes represent the patents and edges indicate the relationship between patents.

When building a patent graph, the root or a higher ranking node (a parent) might has relationship with several lower ranking nodes (children), and directed edges should be draw from the parent to children, note that there might exist children who share the same keyphrases, see Fig. 1(a), node 13, 15 and 20 all have the same keyphrase **Kx**. Correspondingly, due to our principle we should establish sub-relationship between the children nodes (**e1**, **e2** and **e3**). Therefore some elevation is redundant, the original lowest ranking node (20) will probably gain the most promotion, for all the other nodes will elevate it once, apparently it is unreasonable. Given this we prune the e1, e2 and e3 in the graph. A point worth mentioning is that such pruning is not always perfectly justifiable when the shared keyphrases among children are not owned by their parent. See Fig. 1(b), node 14 and 17 share the same keyphrase **Kg**, they have a relatively independent relationship (**e4**) beyond their parent, so the elevation of 17 by means of 14 is not totally affected by node 2. Things will become more complicated with multilevel nodes, considering this will happen less commonly than that in Fig. 1(a) and in order to adopt a general pruning method we neglect this case.

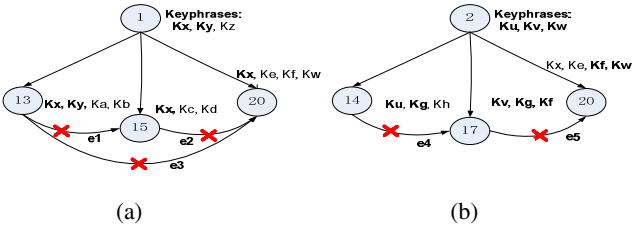


Fig. 1. An example of patent graph; node label denotes the ranking

In our system, we exploit the Breadth First Search method to traverse the graph, whose traversal paths form a shortest path tree, and the redundant edges are pruned.

Based on the analysis above, we develop our reranking formula with the similar idea of PageRank [16]:

$$\text{PatentRank}(\text{patent}_{\text{level-1}}) = \begin{cases} \beta \cdot \frac{\frac{c}{n}\frac{c}{m} \cdot \text{Score}(\text{root})}{2\sqrt{k}} & \text{if level} = 1 \\ \beta \cdot \frac{\frac{c}{n}\frac{c}{m} \cdot \text{PatentRank}(\text{parent}_{\text{level}})}{2\sqrt{k}} & \text{if level} > 1 \end{cases} \quad (2)$$

where

k: is the children number of a parent

n, m, c: denote the keyphrases number of parent, child and that they shared

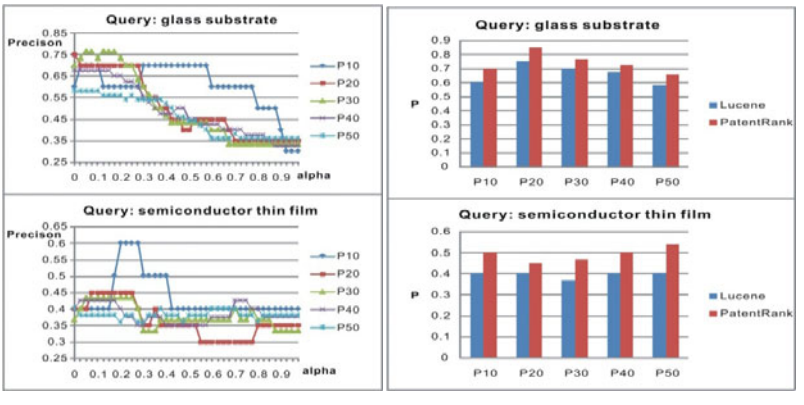
β: is an adjusting parameter, its range is (0, 1]

level: the node's level in the shortest path tree, the root's level is 0

In the formula, **c/n** and **c/m** represent the intimacy between the parent and child; **k** in the denominator indicates the score of parent is shared by its k children, while the square value is to slow down the decaying, given the root might has a myriad of children; in the denominator there is a constant 2, which denotes children could inherit half of its parent's PatentRank score, this idea is borrowed from genetics.

4 Evaluation

To conduct the experiments, we choose 2,000 patents in photovoltaic area and predefine six query expressions. Then we ask 10 human experts to tag each patent so as to identify whether it is relevant to the predefined queries, accordingly we could figure out the answer set to those queries. Besides, we ask the human experts to extract keyphrases from about 500 patents manually, which are used as training set to extract other patents' keyphrases by means of Maui.



(a) (b)

Fig. 2. (a) The precision of query expression "glass substrate" and "semiconductor thin film" when combing IPC with different weights. **(b)** Precision comparison between Lucene and PatentRank.

Our experiment begins with indexing the patents and IPCs by Lucene, and then we execute the query expressions on that index with varying **α** in Formula(1), next we calculate the precision according to the answer set, given the length of the paper, we show only two result figures of our experiment in Fig. 2(a). From the results, we find that when the value of **α** is between 0.15 and 0.3, the precision of search results are on or proximity to their maximums although there are some noises owing to the patent tagging or keyphrases extraction. Note that when**α**= 0, the y-coordinate value denotes the pure Lucene's (without combing IPC) precision.

According to our experiment we typically set **β** = 0.1, Fig. 2(b) are two examples show the comparison between our system and the pure Lucene system (α= 0, β= 0, no keyphrases field and no boost in title field). From the figures we could find precision is improved is our system substantially.

5 Conclusion and Future Work

In this paper we propose a novel approach to patent oriented semantic search, this approach is based on the Lucene search engine, but we introduce IPC in its scoring system which makes the query more understandable by the computer; we also promote the weight of certain field in patent document considering their contribution to represent or identify the document; lastly we discover the relationship between the highly relevant patent and other ones, and upgrade the ranking of the patents that might interest the user. The experiments have proved the validity of our approach.

In the future we will improve the scoring process for IPC documents and make it more effective.

Acknowledgments. This research is supported by National Natural Science Foundation of China (Grant No. 61003100 and No. 60972011) and Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20100002120018 and No. 20100002110033).

Reference

1. Guha, R., McCool, R., Miller, E.: Semantic Search. In: Proceedings of the 12th International Conference on World Wide Web, Budapest, Hungary, May 20-24 (2003)
2. Mangold, C.: A survey and Classification of Semantic Search Approaches. International Journal of Metadata, Semantics and Ontologies 2(1), 23–34 (2007)
3. Dong, H., Hussain, F.K., Chang, E.: A Survey in Semantic Search Technologies. In: 2nd IEEE International Conference on Digital Ecosystems and Technologies, pp. 403–408 (2008)
4. Delbru, R., Toupikov, N., Catasta, M., Tummarello, G.: A Node Indexing Scheme for Web Entity Retrieval. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6089, pp. 240–256. Springer, Heidelberg (2010)
5. Bao, Z., Lu, J., Ling, T.W., Chen, B.: Towards an Effective XML Keyword Search. IEEE Transactions on Knowledge and Data Engineering 22(8), 1077–1092 (2010)

6. Shah, U., Finin, T., Joshi, A., Cost, R.S., Matfield, J.: Information Retrieval on the Semantic Web. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, McLean, Virginia, USA, November 04-09 (2002)
7. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V., Sachs, J.: Swoogle: A Search and Meta Data Engine for the Semantic Web. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management (CIKM 2004), Washington D.C., USA, pp. 652–659 (2004)
8. Stojanovic, N., Studer, R., Stojanovic, L.: An Approach for the Ranking of Query Results in the Semantic Web. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 500–516. Springer, Heidelberg (2003)
9. Anyanwu, K., Maduko, A., Sheth, A.: SemRank: Ranking Complex Relationship Search Results on the Semantic Web. In: Proceedings of the 14th International World Wide Web Conference. ACM Press (May 2005)
10. Bamba, B., Mukherjea, S.: Utilizing Resource Importance for Ranking Semantic Web Query Results. In: Bussler, C.J., Tannen, V., Fundulaki, I. (eds.) SWDB 2004. LNCS, vol. 3372, pp. 185–198. Springer, Heidelberg (2005)
11. Price, S., Nielsen, M.L., Delcambre, L.M.L., Vedsted, P.: Semantic Components Enhance Retrieval of Domain-Specific Documents. In: 16th ACM Conference on Information and Knowledge Management, pp. 429–438. ACM Press, New York (2007)
12. Sharma, S.: Information Retrieval in Domain Specific Search Engine with Machine Learning Approaches. World Academy of Science, Engineering and Technology 42 (2008)
13. Apache Lucene, `http://lucene.apache.org/`
14. Maui-indexer, `http://code.google.com/p/maui-indexer/`
15. KEA, `http://www.nzdl.org/Kea/`
16. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project (1998)

Fast Growing Self Organizing Map for Text Clustering

Sumith Matharage[1], Damminda Alahakoon[1], Jayantha Rajapakse[2], and Pin Huang[1]

[1] Clayton School of Information Technology, Monash University, Australia
{sumith.matharage,damminda.alahakoon}@monash.edu,
phua13@student.monash.edu
[2] School of Information Technology, Monash Univeristy, Malaysia
jayantha.rajapakse@monash.edu

Abstract. This paper presents an integration of a novel document vector representation technique and a novel Growing Self Organizing Process. In this new approach, documents are represented as a low dimensional vector, which is composed of the indices and weights derived from the keywords of the document. An index based similarity calculation method is employed on this low dimensional feature space and the growing self organizing process is modified to comply with the new feature representation model. The initial experiments show that this novel integration outperforms the state-of-the-art Self Organizing Map based techniques of text clustering in terms of its efficiency while preserving the same accuracy level.

Keywords: GSOM, Fast Text Clustering, Document Representation.

1 Introduction

With the rapid growth of the internet and the World Wide Web, availability of text data has massively increased over the recent years. There has been much interest in developing new text mining techniques to convert this tremendous amount of electronic data into useful information. There have been different techniques developed to explore, organize and navigate massive collections of text data over the years, but there is still for improvement in the existing techniques' capabilities to handle the increasing volumes of textual data.

Text Clustering is one of the most promising text mining techniques, which groups collection of documents based on their similarity. Moreover, it identifies inherent groupings of textual information by producing a set of clusters, which exhibits high level of intra-cluster similarity and low inter-cluster similarity [1].

Text clustering has received special attention from researchers in the past decades [2, 3]. Among many of the different text clustering techniques, the Self Organizing Map (SOM) [4] and many of its variants have shown great promise [5, 6]. But, many of these algorithms do not perform efficiently for large volume of text data. This performance drawback occurs due to the very frequent similarity calculations that become necessary in the high dimensional feature space and thus becoming a critical issue when handling large volumes of text.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 406–415, 2011.
© Springer-Verlag Berlin Heidelberg 2011

There has been different techniques introduced to overcome these limitations, but still there is a significant gap between the current techniques and what is required. This paper introduces a novel integration of document vector representation and a modified growing self organizing process to cater for this new document representation, which leads to a more efficient text clustering algorithm, while preserving the same accuracy of the results. The initial experiments have shown that this novel algorithm have capabilities to bridge the efficiency gap in the existing text clustering techniques.

The rest of the paper is organized as follows. Section 2 describes the related work on the document representation and SOM based text clustering techniques. Section 3 describes the new document feature selection algorithm followed by the Fast Growing Self Organizing Map algorithm in section 4. Section 5 describes the experimental results and related discussion. Finally, section 6 concludes the findings together with the future work.

2 Related Work

2.1 Document Vector Representation

Text documents need to be converted into a numerical representation in order to be fed into the existing clustering algorithms. Most of the existing clustering algorithms use Vector Space Model (VSM) to represent the documents. In VSM, each document is represented with a multi dimensional vector, in which the dimensionality corresponds to the number of words or the terms in the document collection and value (the weight) represent the importance of the particular term in the document. The main drawback of this technique is that, the number of terms extracted from a document corpus is comparatively large resulting in high dimensional sparse vectors. This has a negative effect on the efficiency of the clustering algorithm.

To overcome this different feature selection and dimensionality reduction mechanisms have been introduced. A systematic comparison of these dimensionality reduction techniques has been presented in [7]. In terms of feature selection, [8] has proven that each document can be represented using a few words. If we can represent a document with a fewer number of words this will remove the sparsity of the input vector, resulting in low dimensional vectors.

To overcome the above issues, an index based document representation technique for efficient text clustering is proposed in FastSOM [9]. In FastSOM, a document is represented as a collection of indexes of the keywords present only in the document instead of the high dimensional feature vector which is constructed from the keywords extracted from the entire document set. Since a single document only contains a smaller amount of terms from the entire feature space, this will result in very low dimensional input vectors. The experiments have proven that the resulting low dimensional vectors significantly increase the efficiency of the clustering algorithm.

Term weighting is another important technique when converting documents into its numerical representation. Although there have been different term weighting techniques proposed, in [8] it is shown that the term frequency itself will be

sufficient rather using complex calculations which will increase the computation time. But, the FastSOM [9] doesn't use a term weighting technique, rather it uses only whether a particular term is present in the document. But in general, if a particular term is more frequent in a document, it contributes more to the meaning of the document than the less frequent terms. Therefore incorporating a term weighting technique would increase the usage of the FastSOM algorithm.

Based on the above findings, a novel document representation is presented in this paper by the combining the above mentioned advantages to overcome the limitations of the existing techniques. The detailed document representation algorithm is presented in Section 3.

2.2 SOM Based Text Clustering Techniques

The Self Organizing Map (SOM) is a mostly distinguished unsupervised neural network based clustering technique which resembles the self organizing characteristics of the human brain. SOM maps a high dimensional input space into a lower dimensional output space while preserving the topological properties of the input space. SOM has been extensively used across different disciplines, and has shown impressive results. Moreover, in text clustering research it has been proven as one of the best text clustering and learning algorithms [10].

SOM consists of a collection of neurons, which are arranged in a two dimensional rectangular or hexagonal grid. Each neuron consists of a weight vector, which has the same dimensionality as the input patterns. During the training process, similarity between the input patterns and the weight vectors are calculated and the winner (the neuron with the closest weight vector to the input pattern) is selected and the weight vectors of the winner and its neighborhood is adapted towards the input vector.

There have been different variations of the SOM introduced to improve the usefulness for data clustering applications. Among those, different algorithms such as, incremental growing grid [11], growing grid [12], and Growing SOM (GSOM) [13] have been proposed to address the shortcomings of SOM's pre-fixed architecture. Among those, GSOM has been widely used in many of the applications across multiple disciplines. GSOM starts with a small map (mostly with a four nodes map) and adds neurons as required during the training phase, resulting a more efficient algorithm.

More specifically, different variations of SOM and GSOM have been widely used in text mining applications. WEBSOM [14], GHSOM [15] and GSOM [13] are a few of the mostly used algorithms in the text clustering domain. In the next section, we propose a novel algorithm based on the key features of SOM and GSOM with the capability to support new document representation technique presented in Section 4.

3 Document Vector Representation

The detailed novel document representation technique is presented in this section. In our approach, Term frequency is used as the term weighting technique. Each of the documents is represented as a map of <index, weight> pairs corresponding to the keywords present in the document.

doc = *Map* (<*index*, *weight*>)

Term frequency tf_{ij} of term t_i in document d_j is calculated as,

$$tf_{ij} = n_i / \sum_{k=0}^{N} n_k \quad (1)$$

where n_i is the number of occurrences of term t_i and *N* is the number of keywords in the document d_j.

The document vector representation algorithm is described below. (The above notations *doc* and tf_{ij} have the same meaning in the following algorithms)

Algorithm 1. Document Vector Representation

Input : *documentCollection*– collection of input text data
Output : *keywordSet* – represent the complete keyword set
docmentMap - Final representation of the document map
Algorithm :
for (document d_j in *documentCollection*)
tokenSet= *tokenize*(d_j)
for(token t_i in *tokenSet*)
if (t_i is not in *keywordSet*)
add t_i into *keywordSet*
calculate $tf_{i,j}$
add index i and $tf_{i,j}$ pair into doc_j
add doc_j into *docmentMap*

tokenize(document) – This function tokenizes the content of the given document. Also, further preprocessing is carried out to remove the stop words, stem terms and to extract important terms based on the given lower and upper threshold values.

4 Fast Growing Self Organizing Map (FastGSOM) Algorithm

FastGSOM algorithm is a faithful variation of GSOM to support the efficient text clustering. There are three main modifications included in this novel approach.

1. The input document's vectors and the neuron's weight vectors are represented as vectors with different dimensionalities, because of the novel document representation technique introduced. The neurons weights are represented as a high dimensional vector similar to that of the GSOM, while the input document vectors having a lower dimensionality corresponds to the number of different terms present in that document. A new similarity calculation method is employed to cater for this new representation.
2. Weight adaptation of the neurons is modified, to only adapt the weights of the indices in the input document vectors. In addition, the term frequencies are used to

update the weight instead of the error calculated between the input and the neuron in the GSOM.

3. Growing criteria of the GSOM is also modified. The automatic growth of the network is no longer dependent on the accumulated error, but depends on whether the existing neurons are good enough to represent the current input using the similarity threshold. If the existing neurons don't have the required similarity level, new neurons are added to the network.

The detailed algorithm is explained in the following section. The algorithm consists of 3 phases, namely, Initialization, Training and Smoothing phases.

4.1 Initialization Phase

A network is initialized with four nodes. Each of these four nodes contain a weight vector, that has a dimensionality equal to the total number of features extracted from the entire document collection. Each of these weights is initialized as below.

$$w = rand(0,1)/s \tag{2}$$

where w – is the weight value, $rand(0,1)$ function generates a random value in the range of 0 and 1 and s is the initialization seed. *Similarity Threshold* (*ST*), which determines the growth of the network is initialized as,

$$ST = SF \times \log(D) \tag{3}$$

where *SF* is the Spread Factor and *D* is the dimensionality of the neurons weight vector.

4.2 Training Phase

During the training phase, the input document collection *docmentMap* is repeatedly fed into the algorithm for a given number of iterations. The algorithm is explained in detail below.

Algorithm 2. FastGSOM Training Algorithm

```
Input :docmentMap, noOfIterations
Algorithm :
    for (iteration i in noOfIterations)
        for (document doc_j in docmentMap)
            Neuron winner = CalculateSimilarity(doc_j)
            if (winner->similarity<ST)
                GrowNetwork(winner)
            UpdateWeights(winner,doc_j)
```

CaclculateSimilarity, GrowNetwork and UpdateWeightsalgorihms are described below.

The Similarity Calculation Algorithm describes the index based similarity calculation and the modified winner finding algorithm.

Algorithm 3. Similarity Calculation Algorithm

Input : *doc* - represent an input document
Output : *winner* – most similar neuron to the input *doc*
Algorithm :
winner– to keep track of the current winning neuron
maxSimilarity = 0 – to keep track of the current highest similarity
for (neuron neu_i in *doc->neuronSet)*
Similarity = 0;
for (<index, weight> *item* in *doc*)
Similarity +=neu_i[*index*]
if*(Similarity > maxSimilarity*)
winner **=** neu_i
maxSimilarity = Similarity
return *winner*

Note : neu_i[*index*] - return the weight value of neuron neu_i at index *index*

Weight updating algorithm describes the index based weight adaptation algorithm. This is used to update the winner's weights and its neighborhood neurons weights.

Algorithm 4. Weight Updating Algorithm

Input :*neuron*, *doc*
Algorithm :
for (index i in *neuron ->weights*)
if (i is in *doc ->indexes*)
neuron[*i*] += (1 - *neuron* [*i*]) * *LR* * *doc*[*i*] * *distanceFactor*
else
neuron [*i*] -= (*neuron* [*i*]- 0) * *FR** *distanceFactor*

Note - *neuron* [*i*] return the weight value of the index *i, doc*[*i*] returns the weight value corresponding to the index *I* of the input document. LR is the learning rate and FR is the forgetting rate. *distanceFactor* returns a value based on the following Gaussian distribution function.

$$distanceFactor = e^{-((dx^2+dy^2)/r^2)} \quad (4)$$

where dx – x distance between *winner* and $neighbor_i$, dy – y distance between *winner* and $neighbor_i$ and r – learning radius which is taken as a parameter from the user. The value of the distance factor is 1 for the winner and it decreases as the neuron goes away from the winner.

Network growth and weight initialization of the new nodes is something similar to that of GSOM. The algorithm checks whether the top, bottom, left and right neighbors are already present and if not new neurons are added to complete the winner's neighborhood. The weights of the newly created neurons are initialized based on its neighborhood. A detailed weight initialization algorithm is not presented in this paper due to the space limitations, but is exactly similar to that of GSOM [13].

4.3 Smoothing Phase

The Smoothing phase is exactly similar to that of the training phase except for the following differences.

1. No new nodes will be added to the network during the smoothing phase, only the weight values of the neurons are updated.
2. A small Learning Rate and a small neighborhood radius is used.

5 Experimental Results and Discussion

A set of experiments were conducted on the Reuters-21578 "ApteMod" corpus for text categorization. ApteMod is a collection of 10,788 documents from the Reuters financial newswire service, partitioned into a training set with 7769 documents and a test set with 3019 documents. A subset of this data set is used to analyse the different aspects of the FastGSOM algorithm in text clustering tasks.

5.1 Comparative Analysis of Accuracy and Efficiency of FastGSOM

Experiment 1: Comparing the accuracy of FastGSOM with SOM, GSOM
This experiment was conducted to measure the accuracy and the efficiency of the algorithm. The results are compared with that of the SOM and GSOM and are presented below. A subset of documents belonging to the above mentioned dataset is used. In detail, 50 documents from each of the six categories namely, acquisition, trade, jobs, earnings, interest and crude are used in this experiment. The resulting map structures are illustrated in Fig. 1.

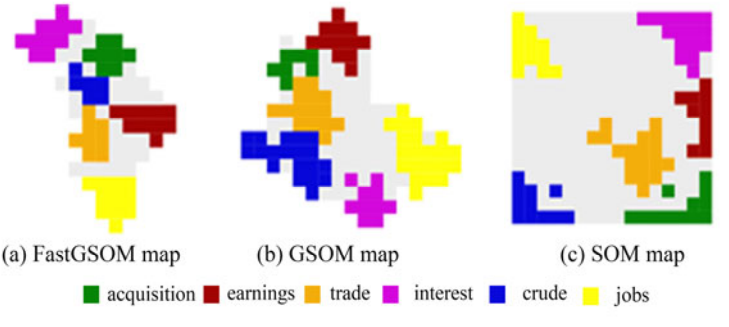


Fig. 1. Resulting Map Structures

The accuracy of the cluster results are calculated using the existing Reuter categorisation information as the basis. Precision, Recall and F-measure values are used as the accuracy measurements.

The Precision P and Recall R of a cluster *j* with respect to a class i are defined as,

$$P = Precision(i,j) = M_{i,j}/M_j \tag{5}$$

$$R = Recall(i,j) = M_{i,j}/M_i \tag{6}$$

where $M_{i,j}$ is the number of members of class i in cluster *j*, M_j is the number of members of cluster j, and M_i is the number of members of the class i. The F-measure of a class i is defined as,

$$F(i) = 2\,P\,R/(P+R) \tag{7}$$

The resulted values are summarized in the table 1.

Table 1. Calculated Precision, Recall and F-Measure values for individual classes

Class	Precision			Recall			F- Measure		
	SOM	GSOM	Fast GSOM	SOM	GSOM	Fast GSOM	SOM	GSOM	Fast GSOM
acquisitions	0.83	0.84	0.83	0.92	0.90	0.90	0.87	0.87	0.86
trade	0.79	0.79	0.80	0.88	0.82	0.83	0.83	0.80	0.81
jobs	0.92	0.88	0.86	0.90	0.92	0.90	0.91	0.90	0.88
earnings	0.85	0.84	0.82	0.86	0.84	0.88	0.85	0.84	0.85
interest	0.86	0.86	0.81	0.88	0.88	0.86	0.87	0.87	0.83
crude	0.83	0.86	0.85	0.84	0.82	0.84	0.83	0.84	0.84

Experiment 2: Comparing the efficiency of FastGSOM with SOM and GSOM
This experiment was conducted to compare the efficiency of the algorithm with that of the SOM and GSOM. Different subsets of the same six classes are selected and the processing time was recorded. In addition, the computation times were also recorded separately for different spread factor values for the same document collection. The results are illustrated in Fig. 2.

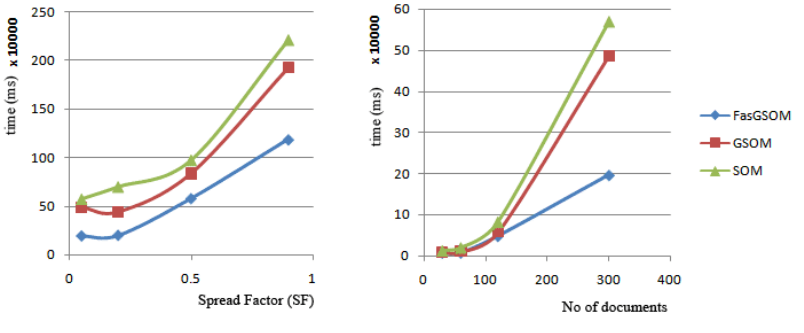


Fig. 2. Comparison of efficiency (a) Time Vs Spread Factor (b) Time Vs No of Documents

From the above results it is evident that the FastGSOM preserves the same accuracy as SOM and GSOM while giving a performance advantage over them. This performance advantage is more significant in low granularity (high detailed) maps and when the number of documents in the document collection is large.

5.2 Theoretical Analysis of the Runtime Complexity of the Algorithm

The theoretical aspects of the runtime complexity of the SOM, GSOM and FastGSOM algorithms are described in this section together with some evidence from the experimental results.

In SOM based algorithms, similarity calculation is more frequent, and it happens in the n dimensional feature space, where n is dimensionality of the input vectors. Therefore run time complexity of Similarity calculation is O (n).

This similarity calculation is performed, (k * m * N) times where k is the number of neurons, m is the number of training iterations and N is the number of documents. Therefore, the complete runtime complexity of the SOM algorithm is O (n * k * m * N). In the GSOM algorithm, because of the initial small size of the network and it will only grow the neurons as necessary, $k_{GSOM} < k_{SOM}$ resulting a more efficient calculation with a low computational time.

Since the FastGSOM is also based on the growing self organizing process, it also has above mentioned performance advantage. But in addition, because of the novel feature representation technique introduced, a dimension of a document becomes very small compared to that of the complete feature set. As such, $n_{FastGSOM} < n_{GSOM}$ resulting even better efficiency in the algorithm. Based on the above theoretical aspects, we can summarized that, $Efficiency_{SOM} < Efficiency_{GSOM} < Efficiency_{FastGSOM}$. Experimental results have already proved this theoretical explanation.

6 Conclusions and Future Research

In this paper, we presented a novel growing self organizing map based algorithm to facilitate efficient text clustering. The high efficiency was obtained by using the novel method of index based document vector representation and modified growing self organizing process based on index based similarity calculation introduced in this paper.

The initial experiments were conducted to test accuracy, and efficiency of the algorithm in detail, using a subset of Reuters-21578 "ApteMod" corpus, and the results have proved the above mentioned advantages of the algorithm.

There are a number of future research directions to extend and improve the work presented here. We are currently working on building a cognition based incremental text clustering model using the efficiency and the hierarchical capabilities of the FastGSOM algorithm,. Also, there is some room to analyze other aspects of the algorithm with its parameters, and fine-tune the algorithm to obtain even better results.

References

1. Rigouste, L., Cappé, O., Yvon, F.: Inference and evaluation of the multinomial mixture model for text clustering. Information Processing & Management 43(5), 1260–1280 (2007)
2. Aliguliyev, R.M.: Clustering of document collection-A weighting approach. Expert Systems with Applications 36(4), 7904–7916 (2009)
3. Saraçoglu, R.I., Tütüncü, K., Allahverdi, N.: A new approach on search for similar documents with multiple categories using fuzzy clustering. Expert Systems with Applications 34(4), 2545–2554 (2008)
4. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological cybernetics 43(1), 59–69 (1982)
5. Chow, T.W.S., Zhang, H., Rahman, M.: A new document representation using term frequency and vectorized graph connectionists with application to document retrieval. Expert Systems with Applications 36(10), 12023–12035 (2009)
6. Hung, C., Chi, Y.L., Chen, T.Y.: An attentive self-organizing neural model for text mining. Expert Systems with Applications 36(3), 7064–7071 (2009)
7. Tang, B., Shepherd, M.A., Heywood, M.I., Luo, X.: Comparing Dimension Reduction Techniques for Document Clustering. In: Kégl, B., Lee, H.-H. (eds.) Canadian AI 2005. LNCS (LNAI), vol. 3501, pp. 292–296. Springer, Heidelberg (2005)
8. Sinka, M.P., Corne, D.W.: The BankSearch web document dataset: investigating unsupervised clustering and category similarity. Journal of Network and Computer Applications 28(2), 129–146 (2005)
9. Liu, Y., Wu, C., Liu, M.: Research of fast SOM clustering for text information. Expert Systems with Applications (2011)
10. Isa, D., Kallimani, V., Lee, L.H.: Using the self organizing map for clustering of text documents. Expert Systems with Applications 36(5), 9584–9591 (2009)
11. Blackmore, J., Miikkulainen, R.: Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map. IEEE (1993)
12. Fritzke, B.: Growing Grid - a self-organizing network with constant neighborhood range and adaptation strength. Neural Processing Letters 2, 9–13 (1995)
13. Alahakoon, D., Halgamuge, S.K., Srinivasan, B.: Dynamic self-organizing maps with controlled growth for knowledge discovery. IEEE Transactions on Neural Networks 11(3), 601–614 (2000)
14. Kohonen, T., et al.: Self organizing of a massive document collection. IEEE Transactions on Neural Networks 11(3), 574–585 (2000)
15. Rauber, A., Merkl, D., Dittenbach, M.: The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. IEEE Transactions on Neural Networks 13(6), 1331–1341 (2002)

News Thread Extraction Based on Topical N-Gram Model with a Background Distribution

Zehua Yan and Fang Li

Department of Computer Science and Engineering,
Shanghai Jiao Tong University
{yanzehua,fli}@sjtu.edu.cn
http://lt-lab.sjtu.edu.cn

Abstract. Automatic thread extraction for news events can help people know different aspects of a news event. In this paper, we present a method of extraction using a topical N-gram model with a background distribution (TNB). Unlike most topic models, such as Latent Dirichlet Allocation (LDA), which relies on the bag-of-words assumption, our model treats words in their textual order. Each news report is represented as a combination of a background distribution over the corpus and a mixture distribution over hidden news threads. Thus our model can model "presidential election" of different years as a background phrase and "Obama wins" as a thread for event "2008 USA presidential election". We apply our method on two different corpora. Evaluation based on human judgment shows that the model can generate meaningful and interpretable threads from a news corpus.

Keywords: news thread, LDA, N-gram, background distribution.

1 Introduction

News events happen every day in the real world, and news reports describe different aspects of the events. For example, when an earthquake occurs, news reports will report the damage caused, the actions taken by the government, the aid from the international world, and other things related to the earthquake. News threads represent these different aspects of an event.

Topic models, such as Latent Dirichlet Allocation (LDA) [1] can extract latent topics from a large corpus based on the bag-of-words assumption. Actually news reports are sets of semantic units represented by words or phrases. N-gram phrases are meaningful to represent these semantic units. For example, "Bush Government" and "Security Council" in table 1 are two news threads for the "Iran nuclear program" event. They capture two aspects of the meaning of the event reports. Our task is to automatically extract news threads from news reports.

Reports of a news event or a topic discuss the same event or the same topic and share some common words. Based on the analysis of LDA results, we find that such common words represent the background of the event. We then assume each

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 416–424, 2011.
© Springer-Verlag Berlin Heidelberg 2011

news report is represented by a combination of (a) a background distribution over the corpus, (b) a mixture distribution over hidden news threads.

In this paper, we use a topical n-gram model with a background distribution (TNB) to extract news threads from a news event corpus. It is an extension of the LDA model with word order and a background distribution. In the following, our model will be introduced, then experiments described and results given.

Table 1. Threads and news titles for news event "Iran nuclear program"

Event corpus	Thread	News report titles
Iran Nuclear Program	the Security Council	Options for the Security Council
		Iran ends cooperation with IAEA
		Iran likely to face Security Council
	the Bush government	Rice: Iran can have nuclear energy, not arms
		Bush plans strike on Iran's nuclear sites
		Iran Details Nuclear Ambitions

2 Related Work

In [2]'s work, news event threading is defined as the process of recognizing events and their dependencies. They proposed an event model to capture the rich structure of events and their dependencies in a news topic. Features such as temporal locality of stories and time-ordering are used to capture events.

[3] proposed a probabilistic model that accounts for both general and specific aspects of documents. The model extends LDA by introducing a specific aspect distribution and a background distribution. In this paper, each document is represented as a combination of (a) a background distribution over common words, (b) a mixture distribution over general topics, and (c) a distribution over words that are treated as being specific to the documents. The model has been applied in information retrieval and showed that it can match documents both at a general level and at specific word level. Similarly, [4] proposed an entity-aspect model with a background distribution; the model can automatically generate summary templates from given collections of summary articles.

Word order and phrases are often critical to capture the latent meaning of text. Much work has been done on probabilistic generation models with word order influence. [5] develops a bigram topic model on the basis of a hierarchical Dirichlet language model [6], by incorporating the concept of topic into bigrams. In this model, word choice is always affected by the previous word.

[7] proposed an LDA collocation model (LDACOL). Words can be generated from the original topic distribution or the distribution in relation to the previous word. A new bigram status variable is used to indicate whether to generate a bigram or a unigram. It is more realistic than the bigram topic model which always generates bigrams. However, in the LDA Collocation model, bigrams do not have topics because the second term of a bigram is generated from a distribution conditioned on its previous word only.

Further, [8] extended LDACOL by changing the distribution of previous words into a compound distribution of previous word and topic. In this model, a word has the option to inherit a topic assignment from its previous word if they form a bigram phrase. Whether to form a bigram for two consecutive word tokens depends on their co-occurrence frequency and nearby context.

3 Our Methods

3.1 Motivation

We analyze different news reports, and find that there are three kinds of words in a news report: background words (B), thread words (T) and stop words (S). Background words describe the background of the event. They are shared by reports in the same corpus. Thread words illustrate different aspects of an event. Stops words are meaningless and appear frequently across different corpora.

For example, there are two sentences from a news report of "US presidential election" in table 2. The first sentence talks about "immigration policy" and the second discusses "healthcare". Stop words are labeled with "S" such as "as" and "the". Background words are "presidential" and "election" which appear in both sentences and are labeled with "B". Other words are thread words that are specifically associated with different aspects of the event, such as "immigration" and "healthcare".

Table 2. Two sentences from "US presidential election"

As/S we/S approach the/S 2008 Presidential/B election/B,/S both/S John/B McCain/B and/S Barack/B Obama/B are/S sharpening/T their/S perspectives/B on/S immigration/T policy/B./S
After/S the/S economy/T ,/S US/B healthcare/T is/S the/S biggest/T domestic/T issue/T influencing/B voters/B in/S the/S US/B presidential/B election/B ./S

Also, we note that adjacent words can form a meaningful phrase and provide a clearer meaning, for example, "presidential election" and "domestic issue". Based on the analysis, there are four possible combinations as follows:

1. B+B: Presidential/B election/B
2. B+T: US/B healthcare/T
3. T+B: immigration/T policy/B
4. T+T: domestic/T issue/T

There is no doubt that "B+B" is a background phrase, and the "T+T" is a thread phrase. Both "B+T" and "T+B" are regarded as thread phrases because the phrase contains a thread word. For example, immigration is a thread word and policy is a background word; the phrase "immigration policy" identifies a type of "policy", and should be viewed as a thread phrase.

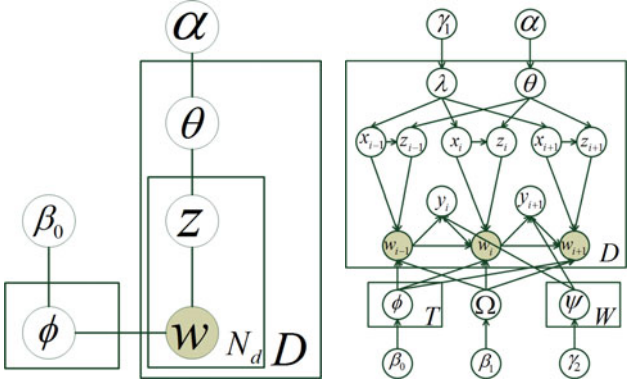
3.2 Topical N-Gram Model with Background Distribution

We now propose our topical n-gram model with a background distribution (TNB) for news reports. Notation used in this paper is listed in table 3. Stop words are identified and removed using a stop word list.

In our model, each news report is represented as a combination of two kinds of multinomial word distribution:

(a) There is a background word distribution Ω with Dirichlet prior parameter β_1, which generates common words across different threads. (b) There are T thread word distributions $\phi_t (1 < t < T)$ with Dirichlet prior parameter β_0. A hidden bigram variable x_i is used to indicate whether a word is generated from the background word distribution or the thread word distribution.

A hidden bigram variable y_i is introduced to indicate whether word w_i can form a phrase with its previous word w_{i-1} or not. Unlike [8], we assume phrase generation is only affected by the the previous word.



(a) LDA (b) TNB

Fig. 1. Graphical model for LDA and TNB

Figure 1 shows graphical models of LDA and TNB. For each word w_i, LDA first draws a topic z_i from the document-topic distribution $p(z|\theta_d)$ and then draws the word from the topic-word distribution $p(w_i|\phi_{z_i})$. TNB has a similar general structure to the LDA model but with additional machinery to identify word w_i's category (background or thread word) and whether it can form a phrase with the previous word w_{i-1}.

For each word w_i, we first sample variable y_i. If $y_i = 0$, w_i is not influenced by w_{i-1}. If $y_i = 1$, w_{i-1} and w_i can form a phrase. As analyzed before, phrases have four possible combinations. There are two situations when $y_i = 1$:

1. if $w_{i-1} \in z_t$, w_i draws either from the thread z_t or the background distribution.

2. if w_{i-1} is a background word, w_i draws from any threads or the background distribution.

Table 3. Notation used in this paper

SYMBOL	DESCRIPTION	SYMBOL	DESCRIPTION
α	Dirichlet prior of θ	β_0	Dirichlet prior of ϕ
β_1	Dirichlet prior of Ω	γ_1	Dirichlet prior of λ
γ_2	Dirichlet prior of σ	T	number of threads
D	number of documents	W	number of unique words
$w_i^{(d)}$	the i^{th} word in document d	$z_i^{(d)}$	the thread associated with i^{th} word in the document d
$y_i^{(d)}$	the bigram status between the $(i-1)^{th}$ word and i^{th} word in the document d	$x_i(d)$	the bigram status indicate the i^{th} word is a background word or topic word
$\theta^{(d)}$	the multinomial distribution of topics w.r.t the document d	ϕ_z	the multinomial distribution of words w.r.t the topic z
Ω	the multinomial distribution of words w.r.t the background	ψ_i	the Bernoulli distribution of status variable $y_i(d)$
λ_i	the Bernoulli distribution of status variable $x_i(d)$		

Second, we sample variable x_i. If $x_i = 1$, w_i is a background word, it is generated from $Multi(\Omega)$. Else it is generated in the same way as LDA.

3.3 Inference

For this model, exact inference over hidden variables is intractable due to the large number of variables and parameters. There are several approximate inference techniques which can be used to solve this problem, such as variational methods [9], Gibbs sampling [10] and expectation propagation [11]. As [12] showed that phrase assignment can be sampled efficiently by Gibbs sampling, Gibbs sampling is adopted for approximate inference in our work.

The conditional probability of w_i given a document d_j can be written as:

$$p(w_i|d_j) = (p(x_i = 0|d_j) \sum_{t=1}^{T} p(w_i|z_i = t, d) + p(x_i = 1|d_j)p'(w)) \times p(w_i|y_i, w_{i-1}) \tag{1}$$

where $p(w_i|z_i = t, d)$ is the thread word distribution and $p'(w)$ is the background word distribution. $p(w_i|y_i, w_{i-1})$ describe the w'_{i-1}sinfluence over w_i.

In Figure 1(b), if $y_i = 0$, the w_i will not be influenced by w_{i-1} and will be generated from the background distribution and thread distribution. Gibbs sampling equations are derived as follows:

$$p(x_i = 0, y_i = 0, z_i = t|w, x_{-i}, z_{-i}, \alpha, \beta_0, \gamma_1, \gamma_2) \propto \frac{N_{d0,-i}+\gamma_1}{N_{d,-i}+2\gamma_1} \times \frac{C^{TD}_{td,-i}+\alpha}{\sum_{t'} C^{TD}_{t'd,-i}+T\alpha} \times \frac{C^{WT}_{wt,-i}+\beta_0}{\sum_{w'} C^{WT}_{w't,-i}+T\beta_0} \times \frac{N_0^{w_{i-1}}+\gamma_2}{N_{w_i-1}+2\gamma_2} \tag{2}$$

$$p(x_i = 1, y_i = 0|w, x_{-i}, z_{-i}, \beta_1, \gamma_1, \gamma_2) \propto \frac{N_{d1,-i}+\gamma_1}{N_{d,-i}+2\gamma_1} \times \frac{C^{W}_{w,-i}+\beta_1}{\sum_{w'} C^{W}_{w',-i}+T\beta_1} \times \frac{N_0^{w_{i-1}}+\gamma_2}{N_{w_i-1}+2\gamma_2} \tag{3}$$

If $y_i = 1$, the w_i can form a phrase with w_{i-1}.

$$p(x_i = 0, y_i = 1, z_i = t | w_{i-1}, z_{i-1} = t, \alpha, \beta_0, \gamma_1, \gamma_2) \propto \frac{N_{d0,-i}+\gamma_1}{N_{d,-i}+2\gamma_1} \times \frac{C^{WT}_{wt,-i}+\beta_0}{\sum_{w'} C^{WT}_{w't,-i}+T\beta_0} \times \frac{N_1^{w_{i-1}}+\gamma_2}{N_{w_i-1}+2\gamma_2} \tag{4}$$

$$p(x_i = 1, y_i = 1 | w_{i-1}, z_{i-1} = t, \alpha, \beta_1, \gamma_1, \gamma_2) \propto \frac{N_{d1,-i}+\gamma_1}{N_{d,-i}+2\gamma_1} \frac{C^{W}_{w,-i}+\beta_1}{\sum_{w'} C^{W}_{w',-i}+T\beta_1} \times \frac{N_1^{w_{i-1}}+\gamma_2}{N_{w_i-1}+2\gamma_2} \tag{5}$$

where the subscript $-i$ stands for the count when word i is removed. N_d is the number of words in document d. N_{d0} stands for the number of thread words in document d, and N_{d1} is the number of background words in document d. $N_{w_{i-1}}$ is the number of words w_{i-1}. $N_0^{w_{i-1}}$ and $N_1^{w_{i-1}}$ is the number of words w_{i-1} which have been drawn from as a unigram or as a part of phrase. C^{WT}_{wt}, C^{W}_{w} are the number of times a word is assigned to a thread t, or to a background distribution respectively.

4 Experiments

4.1 Experimental Settings

Two corpora are used in the experiments. The Chinese news corpus is an event based corpus, which contains 68 event sub-corpora, such as "2007 Nobel prize". The number of news reports in a sub-corpus varies from 100 to 420. Another corpus is the Reuters-21578 financial news corpus. We select five sub corpora from it, they are: "crude", "grain", "interest", "money-fx" and "trade". Each of them contains more than 300 reports which describe many events.

Experiments are run on both corpora with different numbers of threads. The experiments are run with 500 iterations for each case. And we set $\alpha = 50/T$ where T is the number of threads, $\beta_0 = 0.1$, $\beta_1 = 0.1$ and $\gamma_1 = 0.5$, $\gamma_2 = 0.5$ by experience.

The LDA result is used as our baseline. The top three words of LDA are compared with the top three phrases generated by TNB on different corpora at different numbers of threads.

4.2 Evaluation Metrics

There is no golden standard for news thread extraction. Only humans can identify and understand news threads for different news events. The top three phrases of TNB and top three words of LDA are evaluated by voluntary judges on a scale of 0 to 1. Report titles are provided as the basis for judging. Score 1 means the phrase or the word represents the meaning of the title well. Score 0 means the word or the phrase does not capture the meaning of the title. Score 0.5 is between them. The precision of news threads are calculated in the following three formula:

$$top{-}1 = \frac{\sum_t^T score_{t1}}{T} \tag{6}$$

$$top-2 = \frac{\sum_t^T max(score_{t1}, score_{t2})}{T} \tag{7}$$

$$top-3 = \frac{\sum_t^T max(score_{t1}, score_{t2}, score_{t3})}{T} \tag{8}$$

where $score_{ti}$ is the score of the i^{th} word in thread t.

4.3 Results and Analysis

Table 4 and 5 shows the precisions of news thread extraction from the Chinese and Ruters corpus with different numbers of threads. As the number of thread increases, the precision decreases. We analyze both corpora. The Chinese corpus is event-based, the number of 5 or 8 matches its semantic meaning hidden in each event corpus. Twenty threads are adequate to the semantic meanings of the Reuters sub-corpora. The hidden semantics of the corpus dominate the precision and final results.

The precision of TNB is much better than LDA. We give two explanations. Table 7 shows both results extracted from the "2007 Nobel Prize" reports. First, the top LDA words do not consider the background influence, common words such as "Nobel" appearing in the top three words. Such words cannot be regarded as thread words to represent different aspects of an event. In TNB, thread-specific words (such as "Peace") can be extracted and form an n-gram phrase with backgroun word to represent the thread more clearly. The second explanation is that a phrase delivers more clear information than a unigram word. For example, "peace" vs. "Nobel Peace Prize". The top three results of TNB for threads related to the Nobel Peace Prize convey two meanings "Nobel Peace Prize" and "Climate change problem", while people need his knowledge to understand the top three words of LDA.

Table 4. Precision on Chinese corpus

Evaluations	Number of thread			
	5	8	10	12
TNB top-1	72.3%	65.4%	61.5%	60.9%
TNB top-2	85.2%	82.4%	77.7%	75.1%
TNB top-3	90.6%	88.3%	82.9%	81.4%
LDA top-1	43.4%	38.3%	31.9%	30.3%
LDA top-2	51.3%	45.5%	37.5%	36.9%
LDA top-3	58.4%	55.1%	46.9%	43.3%

Table 5. Precision on Reuter corpus

Evaluations	Number of thread		
	20	25	30
TNB top-1	55.2%	44.3%	38.3%
TNB top-2	73.2%	61.1%	57.7%
TNB top-3	81.3%	69.4%	66.3%
LDA top-1	32%	29.5%	28.3%
LDA top-2	41.5%	37%	38.4%
LDA top-3	52%	41.5%	40%

Table 6 lists the background words of five sub-corpora of Reuters news. These sub-corpora are not event-based, The background words still catch many features of each category. For example, words like "wheat", "grain" and "agriculture" are easily identified as background words for the category of grain. The word "say" appears as the top background word for all these sub-corpora. The reason is that reports in the Reuters corpus always reference different peoples' opinions, so the word frequency is really high. Therefore "say" is regarded as a background word.

Table 6. Background words for Reuters corpus

trade	crude	grain	interest	money-fx
say	say	say	say	say
trade	oil	wheat	rate	dollar
japan	company	price	bank	rate
japanese	dlrs	grain	market	blah
official	mln	corn	blah	trade

Table 7. LDA and TNB result for threads of "2007 Nobel prize"

Nobel Peace Prize		**Nobel Economics Prize**	
LDA Result			
Peace	0.032	Nobel	0.041
Nobel	0.025	Sweden	0.035
Climate	0.024	economics	0.029
Gore	0.023	announce	0.027
change	0.019	prize	0.021
president	0.016	date	0.015
committee	0.013	winner	0.014
global	0.013	economist	0.013
TNB Background words			
America	0.015	research	0.013
university	0.013	nobel	0.012
gene	0.011	Prize	0.011
TNB Result			
Nobel Peace Prize	0.033	The Royal Swedish Academy	0.056
Climate change problem	0.032	announce Nobel economics prize	0.052
Climate change	0.018	Swedish kronor	0.038

5 Conclusion

In this paper, we present a topical n-gram model with background distribution (TNB) to extract news threads. The TNB model adds background analysis and the word-order feature to standard LDA. Experiments indicate that our model can extract more interpretable threads than LDA from a news corpus. We also find that the number of threads and the event type can influence the precision of news thread extraction. Experiments show that TNB works well not only on an event-based corpus but also on a topic-based corpus. In the future, we plan to develop a dynamic mechanism to decide a suitable number of threads for different news event types to improve the precision of news thread extraction.

Acknowledgements. This research is supported by the Chinese Natural Science Foundation under Grant Numbers 60873134.The authors thank Mr.Sandy Harris for English improvment and other students for human evaluations in the experiments.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I., Lafferty, J.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
2. Nallapati, R., Feng, A., Peng, F., Allan, J.: Event threading within news topics. In: Proceedings of the Thirteenth ACM International Conference on Information and knowledge Management, pp. 446–453. ACM (2004)
3. Chemudugunta, C., Smyth, P., Steyvers, M.: Modeling General and Specific Aspects of Documents with a Probabilistic Topic Model. In: Advances in Neural Information Processing Systems, pp. 241–242 (2006)
4. Li, P., Jiang, J., Wang, Y.: Generating templates of entity summaries with an entity-aspect model and pattern mining. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 640–649. Association for Computational Linguistics (2010)
5. Wallach, H.M.: Topic modeling: beyond bag-of-words. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 977–984. ACM (2006)
6. MacKay, D.J.C., Peto, L.C.B.: A hierarchical dirichlet language model. Natural language engineering 1(03), 289–308 (1995)
7. Griffiths, T.L., Steyvers, M., Tenenbaum, J.B.: Topics in semantic representation. Psychological Review 114(2), 211 (2007)
8. Wang, X., McCallum, A., Wei, X.: Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In: Seventh IEEE International Conference on Data Mining ICDM 2007, pp. 697–702. IEEE (2007)
9. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Machine learning 37(2), 183–233 (1999)
10. Andrieu, C., De Freitas, N., Doucet, A., Jordan, M.I.: An introduction to mcmc for machine learning. Machine learning 50(1), 5–43 (2003)
11. Minka, T., Lafferty, J.: Expectation-propagation for the generative aspect model. In: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, pp. 352–359. Citeseer (2002)
12. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences of the United States of America 101(suppl. 1), 5228 (2004)

Alleviate the Hypervolume Degeneration Problem of NSGA-II

Fei Peng and Ke Tang

University of Science and Technology of China,
Hefei 230027, Anhui, China
pzbhlp@mail.ustc.edu.cn,
ketang@ustc.edu.cn

Abstract. A number of multiobjective evolutionary algorithms, together with numerous performance measures, have been proposed during past decades. One measure that has been popular recently is the hypervolume measure, which has several theoretical advantages. However, the well-known nondominated sorting genetic algorithm II (NSGA-II) shows a fluctuation or even decline in terms of hypervolume values when applied to many problems. We call it the "hypervolume degeneration problem". In this paper we illustrated the relationship between this problem and the crowding distance selection of NSGA-II, and proposed two methods to solve the problem accordingly. We comprehensively evaluated the new algorithm on four well-known benchmark functions. Empirical results showed that our approach is able to alleviate the hypervolume degeneration problem and also obtain better final solutions.

Keywords: Multiobjective evolutionary optimization, evolutionary algorithms, hypervolume, crowding distance.

1 Introduction

During past decades, a number of multiobjective evolutionary algorithms (MOEAs) have been investigated for solving multiobjective optimization problems (MOPs) [1], [2]. Among them, the nondominated sorting genetic algorithm II (NSGA-II) is regarded as one of the state-of-the-art approaches [3]. Together with the algorithms, various measures have been proposed to assess the performance of algorithms [6]-[8]. One measure that has been popular nowadays is the hypervolume measure, which essentially measures "size of the space covered" [7]. So far, it is the only unary measure that is known to be strictly monotonic with regard to Pareto dominance relation, i.e., whenever a solution set entirely dominates another one, the hypervolume value of the former will be better [9].

However, previous studies showed that, NSGA-II could not obtain solutions with good hypervolume values [5]. By further observation we found that, when applying NSGA-II to many MOPs, the hypervolume value of the solution set obtained in each generation may fluctuate or even decline during the optimization process. We call this problem the "hypervolume degeneration problem" (HDP). HDP may cause confusion about when to stop the algorithm and report solutions, because assigning more computation time to the algorithm can not promise better solutions. Intuitively, one may

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 425–434, 2011.
© Springer-Verlag Berlin Heidelberg 2011

calculate the hypervolume value for the solution set achieved in each generation, and stop the algorithm if the hypervolume value reaches a target value. However, calculating hypervolume for a solution set requires great computational effort, not to mention the computational overhead required for calculating hypervolume in each generation.

In the literature of evolutionary multiobjective optimization (EMO), there have been several approaches for improving NSGA-II, whether in terms of hypervolume vaules or not. Researchers have investigated the effects of assigning different ranks to non-dominated solutions [12]-[15], modifying the dominance relation or objective functions [16]-[18], using different fitness evaluation mechanisms (instead of Pareto dominance) [19], [20], and incorporating user preference into MOEAs [21], [22]. However, the hypervolume degeneration problem has yet been put forward, not to mention any effort on solving the problem. In this paper we illustrated the relationship between HDP and the crowding distance selection of NSGA-II. Then, two methods were proposed to alleviate the problem accordingly. To be specific, a single point hypervolume-based selection is appended to the crowding distance selection probabilistically, in order to achieve a trade-off between preserving diversity and progressing towards the Pareto front. Besides, the crowding distance of a certain solution in NSGA-II is the arithmetic mean of the normalized side lengths of the cuboid defined by its two neighbors [3]. We use the geometric mean of the normalized side lengthes as the crowding distance instead. The new algorithm was namedNSGA-II with geometric mean-based crowding distance selection and single point hypervolume-based selection (NSGA-II-GHV). To verify its effectiveness, we comprehensively evaluated it on four well-known functions.

Compared to existing work on improving NSGA-II, this paper contributes from two aspects. First, from the motivation perspective, we for the first time address the HDP of NSGA-II. Second, from the methodology perspective, we focus on modifying the crowding distance selection of NSGA-II, which is quite different from existing approaches.

The rest of the paper is organized as follows: Section II gives some preliminaries about multiobjective optimization and the hypervolume mesure. Then in Section III we will give a brief introduction to the crowding distance selection, and illustrate the relationship between HDP and the crowding distance selection. Methods for alleviating the HDP are also presented in this section. Experimental study is presented in Section IV. Finally, we draw the conclusion in Section V.

2 Preliminaries

2.1 Dominance Relation and Pareto Optimality

Without loss of generality, we consider a multiobjective minimization problem with m objective functions:

$$\begin{aligned} &\text{minimize } F(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_m(\mathbf{x})) \\ &\text{subject to } x \in \Omega. \end{aligned} \tag{1}$$

where the decision vector $\mathbf{x}$ is a D-dimensional vector. Ω is the *decision (variable) space*, and the *objective space* is R^m.

The dominance relation $\prec$ is generally used to compare two solutions with objective vectors $\mathbf{x} = (x_1, ..., x_m)$ and $\mathbf{y} = (y_1, ..., y_m)$: $\mathbf{x} \prec \mathbf{y}$ *iff* $x_i \leq y_i$ for all $i = 1, ..., m$ and $x_j < y_j$ for at least one index $j \in \{1, ..., m\}$. Otherwise, the relation between the two solutions is called *nondominated*. A solution set S is considered to be a *nondominated set* if all the solutions in S are mutually *nondominated*. The dominance relation $\prec$ can be easily extended to solution sets, i.e., for two solution sets $A, B \subseteq \Omega$, $A \prec B$ *iff* $\forall \mathbf{y} \in B, \exists \mathbf{x} \in A : \mathbf{x} \prec \mathbf{y}$. A solution $\mathbf{x}^\star \in \Omega$ is said to be *Pareto optimal* if there is no solution in decision space that dominates $\mathbf{x}^\star$. The corresponding objective vector $F(\mathbf{x}^\star)$ is then called a *Pareto optimal (objective) vector*. The set of all the Pareto optimal solutions is called *Pareto set*, and the set of their corresponding Pareto optimal vectors is called the *Pareto front*.

2.2 Hypervolume Measure

The Pareto dominance relation $\prec$ only defines a partial order, i.e., there may exist incomparable sets, which could cause difficulties when assessing the performance of algorithms [23]. To tackle this problem, one direction is to define a total ordered performance measure that enables mutually comparable with respect to any two objective vector sets [23]. Specifically, this means that whenever $A \prec B \wedge B \not\prec A$, the measure value of A is strictly better than the measure value of B. So far hypervolume is the only known measure with this property in the field of EMO [23].

The hypervolume measure was first proposed in [7] where it measures the space covered by a solution set. Mathematically, a reference point $\mathbf{x}^r$ should be defined at first. For each solution in a solution set $S = \{\mathbf{x}_i = (x_{i,1}, ..., x_{i,m}) | i = 1, ..., |S|\}$, the volume define by $\mathbf{x}_i$ is $V_i = [x_{i,1}, x_1^r] \times [x_{i,2}, x_2^r] \times ... \times [x_{i,m}, x_m^r]$. All these volumes construct the total volume of S, i.e., $\cup V_i$. Then, the hypervolume of S can be defined as [7], [23]

$$\int_{v \subseteq \cup V_i} \cdots \int 1 \cdot dv. \tag{2}$$

This measure has become more and more popular for assessing the performance of MOEAs nowadays.

3 Alleviate the Hypervolume Degeneration Problem of NSGA-II

3.1 Crowding Distance Selection of NSGA-II

The main feature of NSGA-II is that it employs a fast nondominated sorting and crowding distance calculation procedure for selecting offspring. When conducting selection, taking into account the crowding distance is considered to be beneficial for diversity preservation [3]. It is estimated by calculating the average distance of two adjacent solutions surrounding a particular solution along each objective [3]. As shown in Fig. 1 (a), the crowding distance of solution $\mathbf{x}_i$ is the average side lengths of the cuboid formed by its two adjacent solutions $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i+1}$ (shown with a dashed box). Each

objective value is divided by $f_j^{max} - f_j^{min}, j = 1, ..., m$ for normalization, where f_j^{max} and f_j^{min} stand for the maximum and minimum values of the jth objective function.

NSGA-II continuously accepts nondominated sets with nondominated ranks in ascending order (the lower the better) until the number of accepted solutions exceeds the population size. In this case, the crowding distance selection will be applied to the last accepted nondominated set: Solutions with larger crowding distances will survived.

3.2 Hypervolume Degeneration Problem

When applying NSGA-II to MOPs, we found that the hypervolume value of the population in each generation may fluctuate or even decline. The reason can be illustrated in Fig. 1 (b). $S = \{\mathbf{x}_1, ..., \mathbf{x}_5\}$ is a nondominated set. $\mathbf{y}$ is a new solution that is nondominated with all the points in S. In this situation, the crowding distance selection will be employed on the new set $S \cup \{\mathbf{y}\}$. Apparently the crowding distance of $\mathbf{y}$ is larger than that of $\mathbf{x}_4$. Then, $\mathbf{x}_4$ will be replaced with $\mathbf{y}$ and the resultant new nondominated set will be $S' = \{\mathbf{x}_1, \mathbf{y}, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_5\}$. Hence, the hypervolume of set S' will be the hypervolume of set S minus area of the rectangle A plus area of the rectangle B. Since the area of A can be smaller than that of B, the crowding distance selection may cause a decline in terms of hypervolume values. This problem may even deteriorate in case of more than two objective problems [4].

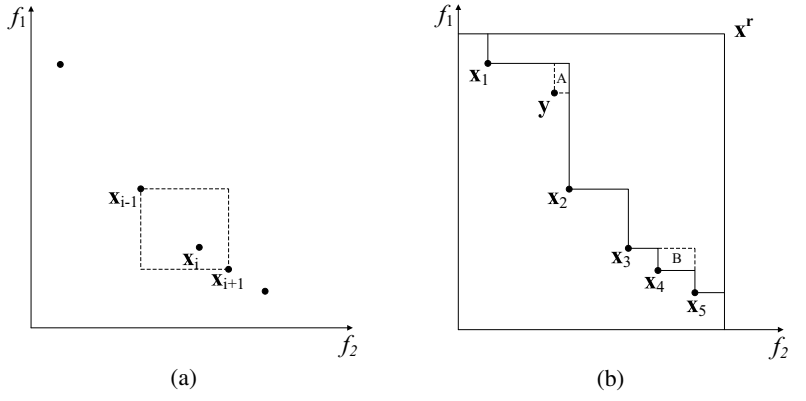


Fig. 1. (a) Crowding distance of calculation. (b) Illustration of the reason for hypervolume degeneration problem in biobjective case.

3.3 NSGA-II with Geometric Mean-Based Crowding Distance Selection and Single Point Hypervolume-Based Selection

We use the original NSGA-II as the basic algorithm, and apply two methods to it in order to alleviate the aforementioned HDP.

Single Point Hypervolume-Based Selection. As illustrated above, the HDP of NSGA-II is due to the fact that, the crowding distance selection always preserves the solutions in sparse area, regardless of how far it is from them to the Pareto front. As a result, solutions which sit close to the Pareto front might be replaced by those ones which are distant from the Pareto font but with larger crowding distances. Consequently, the hypervolume value of the solution set after selection has a possibility to decline. For this reason, preserving some solutions that locate close to Pareto fronts but with small crowding distances may be beneficial. The hypervolume of a solution can indicate the distance between itself and the Pareto front to some extent, and thus can be used for selection. In this paper we simply employed the single point hypervolume-based selection rather than a multiple points based one, because the calculation of hypervolume for multiple points is quite time-consuming. On the other hand, if the algorithm biases too much to those solutions with good hypervolume values, the resultant solution set might assemble together and lose diversity severely. Based on the considerations, a single point hypervolume-based selection is appended to the crowding distance selection probabilistically. In detail, a predefined probability P is given at first. It determines the probability of performing the single point hypervolume-based selection. Then, when the crowding distance selection occurs on a nondominated set S, we modify the procedure as follows:

- Copy S to another set S'.
- Calculate the crowding distances of solutions in S and calculate the hypervolume value of each single solution in S'.
- Sort S and S' according to crowding distance values and single point hypervolume values, respectively.
- Generate a random number r. If $r < P$, choose the solution with largest single point hypervolume value in S' as offspring and remove the solution from S'; otherwise, choose the solution with largest crowding distance in S as offspring and remove the solution from S. Repeat this operation until the number of offspring reaches the limit of the population size.

By applying the new selection, the resultant algorithm would show a trade-off between preserving diversity and progressing closer to the Pareto front. This property is expected to be beneficial for alleviating the HDP while still maintaining diversity to some extent.

Geometric Mean-Based Crowding Distance. As stated in section III-A, the crowding distance of a solution is the arithmetic mean of the side lengths of the cuboid formed by its two adjacent solutions. Since each side length is normalized before conducting the calculation, it is essentially a ratio number, for which geometric mean would be more suitable than arithmetic mean. Moreover, the arithmetic mean can suffer from extremely large or extremely small values, especially the former ones. Thus, the crowding distance selection has an implicit bias to the solutions surrounded by a cuboid with a extremely large length (width) and a extremely small width (length). This bias is usually undesirable. The geometric mean has no such bias, and would be more appropriate for calculating the crowding distance.

4 Experimental Studies

In this section, the effectiveness of the NSGA-II-GHV is empirically evaluated on four well-known benchmark functions chosen from the DTLZ test suite [24]. The problem definitions are given in Table 1. For the four functions, the geometry of the Pareto fronts are totally different, which enables us to fully investigate the performance. We will first compare the hypervolume convergence graphs of NSGA-II-GHV and NSGA-II to verify whether our approach is capable of alleviating the HDP. Further, we will compare the finally obtained Pareto front approximations of our approach with NSGA-II. In all experiments, the objective number was set to three and the dimension of the decision vectors was set to ten.

Table 1. Problem definitions of the test functions. A detailed description can be found in [24].

Problem	Definition
f_1	$f_1(\mathbf{x}) = \frac{1}{2}x_1x_2(1+g(\mathbf{x}))$ $f_2(\mathbf{x}) = \frac{1}{2}x_1(1-x_2)(1+g(\mathbf{x}))$ $f_3(\mathbf{x}) = \frac{1}{2}(1-x_1)(1+g(\mathbf{x}))$ $g(\mathbf{x}) = 100[\lvert\mathbf{x}\rvert - 2 + \sum_{i=3}^{D}((x_i-0.5)^2 - cos(20\pi(x_i-0.5)))]$ $0 \le x_i \le 1, \text{for } i = 1, ..., D, D = 10$
f_2	$f_1(\mathbf{x}) = cos(x_1\pi/2)cos(x_2\pi/2)(1+g(\mathbf{x}))$ $f_2(\mathbf{x}) = cos(x_1\pi/2)sin(x_2\pi/2)(1+g(\mathbf{x}))$ $f_3(\mathbf{x}) = sin(x_1\pi/2)(1+g(\mathbf{x}))$ $g(\mathbf{x}) = 100[\lvert\mathbf{x}\rvert - 2 + \sum_{i=3}^{D}((x_i-0.5)^2 - cos(20\pi(x_i-0.5)))]$ $0 \le x_i \le 1, \text{for } i = 1, ..., D, D = 10$
f_3	$f_1(\mathbf{x}) = cos(\theta_1\pi/2)cos(\theta_2)(1+g(\mathbf{x}))$ $f_2(\mathbf{x}) = cos(\theta_1\pi/2)sin(\theta_2)(1+g(\mathbf{x}))$ $f_3(\mathbf{x}) = sin(\theta_1\pi/2)(1+g(\mathbf{x}))$ $g(\mathbf{x}) = \sum_{i=3}^{D} x_i^{0.1}$ $\theta_1 = x_1, \theta_2 = \frac{\pi}{4(1+g(\mathbf{x}))}(1+2g(\mathbf{x})x_2)$ $0 \le x_i \le 1, \text{for } i = 1, ..., D, D = 10$
f_4	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = x_2$ $f_3(\mathbf{x}) = h(f_1, f_2, g)(1+g(\mathbf{x}))$ $g(\mathbf{x}) = \frac{9}{\lvert\mathbf{x}\rvert-2}\sum_{i=3}^{D} x_i$ $h(f_1, f_2, g) = 3 - \sum_{i=1}^{2}(\frac{f_i}{1+g}(1+sin(3\pi f_i)))$ $0 \le x_i \le 1, \text{for } i = 1, ..., D, D = 10$

4.1 Experimental Settings

All the results presented were obtained by executing 25 independent runs for each experiment. For NSGA-II, we adopted the parameters suggested in the corresponding publications [3]. The population sizes for the two algorithms were set to 300, and the maximum generations were set to 250.

For the single point hypervolume-based selection, two issues need to be figured out in advance. First, the probability P was set to 0.05. Then, the objective values of each

solution were normalized before calculating the hypervolume value. Since solutions could be far away from the Pareto front at the early stage, we simply use the upper and lower bound of each function to employ the normalization. By using relaxation method, the upper and lower bounds of f_1–f_3 were set to 900 and 0 and for f_4 they were set to 30 and -1, respectively. After that, the reference point can be simply chosen at $(1, 1, 1)$.

4.2 Results and Discussions

Figs. 2 (a)–(d) present the evolutionary curves of the two algorithms on the four functions in terms of the hypervolume value of solution set obtained in each generation. For each algorithm, we sorted the 25 runs by the hypervolume values of the final solution sets and picked out the median ones. The corresponding curve was then plotted. Accordingly, the objective values were normalized as discussed in section IV-A and the reference point was also set at $(1, 1, 1)$.

The Pareto front of f_1 is a hyper-plane, while the Pareto front of f_2 is the eighth spherical shell [24]. In this case, NSGA-II showed a fluctuation or decline in hypervolume at the late stage, as showed in Fig. 2 (a) and (b). In most cases NSGA-II fluctuated when it reached a good hypervolume value, which indicates that it might reach a good Pareto front approximation. We also found that most of the solutions are nondominated at this time. Then, the crowding distance selection would play an important part and thus should take the main responsibility for the HDP. On the contrary, NSGA-II-GHV smoothed the fluctuation. Meanwhile, NSGA-II-GHV generally converged faster and finally obtained better Pareto front approximations than NSGA-II.

In Fig. 2 (c), both the two algorithms showed smooth convergence curves. The reason is, the Pareto front of this function is a continuous two-dimensional curve, and thus NSGA-II did not suffer greatly from the HDP as on three-dimensional surfaces. However, NSGA-II generally achieved higher convergence speed.

Finally, in Fig. 2 (d), both the two algorithm suffered from the degeneration problem. The Pareto front of f_4 is a three-dimensional discontinuous surfaces, which leads to great search difficulty. Anyhow, the convergence curve of our approach is generally above the one of NSGA-II.

Below we will further investigate whether NSGA-II-GHV is able to obtain better final solution sets. The hypervolume and inverted generational distance (IGD) [25] were chosen as the performance measures. When calculating hypervolume, we also normalized the objective values and set reference point as mentioned in section II-A. In consequence, the hypervolume values would be quite close to 1, which causes difficulty for demonstration. Hence, we subtracted these values from 1 and presented the mean of the modified values in Table 2. Since a large hypervolume value is considered to indicate a good performance, then the item in Table 2 will indicates good performance when it is small. Two-sided Wilcoxon rank-sum tests [26] with significance level 0.05 have also been conducted based on these values. The one that is significantly better was highlighted in boldface. It can be found that the NSGA-II-GHV outperformed NSGA-II on three out of the four functions, and the difference between the two algorithms is not statistically significant on f_3. Meanwhile, NSGA-II-GHV achieved comparable or superior results than NSGA-II in terms of the IGD values. Hence, the advantage of NSGA-II-GHV has also been verified.

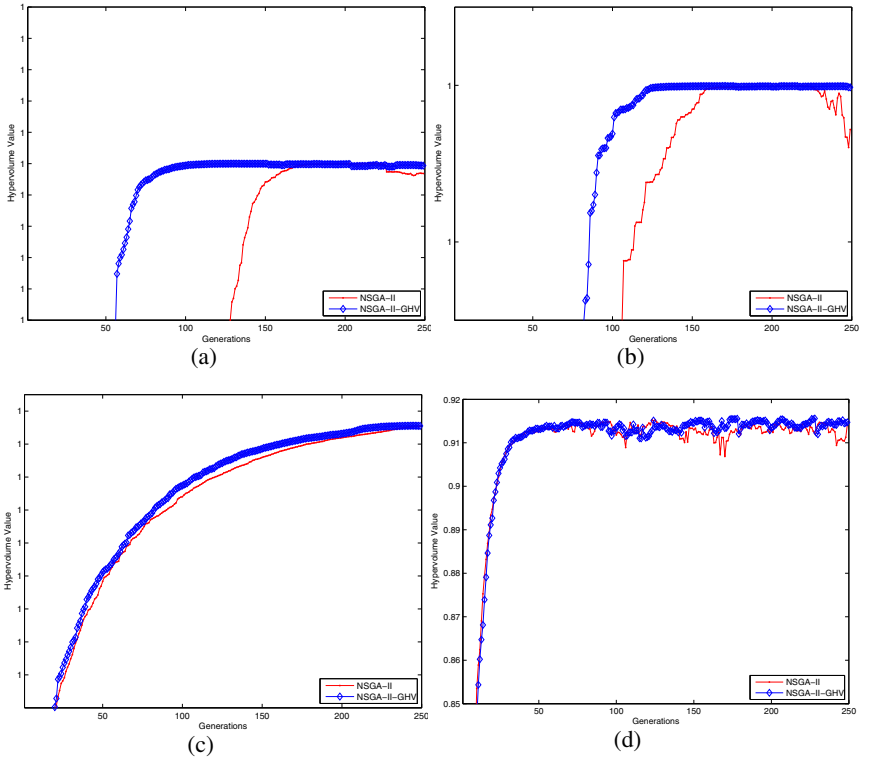


Fig. 2. The hypervolume evolutionary curves of NSGA-II-GHV and NSGA-II on function f_1–f_4

Table 2. Comparison between NSGA-II-GHV and NSGA-II in terms of hypervolume and IGD values

Function	NSGA-II-GHV		NSGA-II	
	hypervolume	IGD	hypervolume	IGD
f_1	$\mathbf{1.54e-09}$	$3.08e-01$	$8.46e-09$	$2.80e-01$
f_2	$\mathbf{8.48e-09}$	$\mathbf{4.36e-02}$	$5.52e-08$	$9.90e-02$
f_3	$1.44-06$	$3.71e-02$	$1.46-06$	$4.68e-02$
f_4	$\mathbf{8.57e-02}$	$1.97e-01$	$8.66e-02$	$1.96e-01$

5 Conclusions

In this paper, the HDP of NSGA-II was identified at first. Then, we illustrated that this problem is due to the fact that the crowding distance selection of NSGA-II always favors the solutions in sparse area regardless of the distances between them and the Pareto front. To solve this problem, a single point hypervolume-based selection was first appended to the crowding distance selection probabilistically to achieve a trade-off between preserving diversity and progressing towards good Pareto fronts. At the same

time, the crowding distance of a solution is the arithmetic mean of side lengths of the cuboid surrounded by its two neighbors. Since the arithmetic mean suffers greatly from extreme values, it will make the crowding distance selection bias towards solutions surrounded by a cuboid with a extremely large length (width) and a extremely small width (length). Therefore, we use a geometric mean instead to remove this bias.

To demonstrate the effectiveness, we comprehensively evaluated the new algorithm on four well-known benchmark functions. Empirical results showed that the proposed methods are capable of alleviating the HDP of NSGA-II. Moreover, the new algorithm also achieved superior or comparable performance in comparison with NSGA-II.

Acknowledgment. This work is partially supported by two National Natural Science Foundation of China grant (No. 60802036 and No. U0835002) and an EPSRC grant (No. GR/T10671/01) on "Market Based Control of Complex Computational Systems."

References

1. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, New York (2001)
2. Coello, C.: Evolutionary multi-objective optimization: A historical view of the field. IEEE Computational Intelligence Magazine 1(1), 28–36 (2006)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
4. Wang, Z., Tang, K., Yao, X.: Multi-objective approaches to optimal testing resource allocation in modular software systems. IEEE Transactions on Reliability 59(3), 563–575 (2010)
5. Nebro, A.J., Luna, F., Alba, E., Dorronsoro, B., Durillo, J.J., Beham, A.: AbYSS: Adapting scatter search to multiobjective optimization. IEEE Transactions on Evolutionary Computation 12(4), 439–457 (2008)
6. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8(2), 173–195 (2000)
7. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)
8. Tan, K., Lee, T., Khor, E.: Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. Artificial Intelligence Review 17(4), 253–290 (2002)
9. Bader, J., Zitzler, E.: HypE: An algorithm for fast hypervolume-based many-objective optimization. Evolutionary Computation 19(1), 45–76 (2011)
10. Ishibuchi, H., Tsukamoto, N., Hitotsuyanagi, Y., Nojima, Y.: Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems. In: 10th Annual Conference on Genetic and Evolutionary Computation (GECCO 2008), pp. 649–656. Morgan Kaufmann (2008)
11. Corne, D., Knowles, J.: Techniques for highly multiobjective optimization: Some nondominated points are better than others. In: 9th Annual Conference on Genetic and Evolutionary Computation (GECCO 2007), pp. 773–780. Morgan Kaufmann (2007)
12. Drechsler, N., Drechsler, R., Becker, B.: Multi-objective Optimisation Based on Relation Favour. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) EMO 2001. LNCS, vol. 1993, pp. 154–166. Springer, Heidelberg (2001)
13. Köppen, M., Yoshida, K.: Substitute Distance Assignments in NSGA-II for Handling Many-Objective Optimization Problems. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 727–741. Springer, Heidelberg (2007)

14. Kukkonen, S., Lampinen, J.: Ranking-dominance and many-objective optimization. In: 2007 IEEE Congress on Evolutionary Computation (CEC 2007), pp. 3983–3990. IEEE Press (2007)
15. Sülflow, A., Drechsler, N., Drechsler, R.: Robust Multi-Objective Optimization in High Dimensional Spaces. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 715–726. Springer, Heidelberg (2007)
16. Sato, H., Aguirre, H.E., Tanaka, K.: Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 5–20. Springer, Heidelberg (2007)
17. Branke, J., Kaußler, T., Schmeck, H.: Guidance in evolutionary multi-objective optimization. Advances in Engineering Software 32(6), 499–507 (2001)
18. Ishibuchi, H., Nojima, Y.: Optimization of Scalarizing Functions Through Evolutionary Multiobjective Optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 51–65. Springer, Heidelberg (2007)
19. Ishibuchi, H., Nojima, Y.: Iterative approach to indicator-based multiobjective optimization. In: 2007 IEEE Congress on Evolutionary Computation (CEC 2007), pp. 3697–3704. IEEE Press, Singapore (2007)
20. Wagner, T., Beume, N., Naujoks, B.: Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 742–756. Springer, Heidelberg (2007)
21. Deb, K., Sundar, J.: Reference point based multi-objective optimization using evolutionary algorithms. In: 8th Annual Conference on Genetic and Evolutionary Computation (GECCO 2006), pp. 635–642. Morgan Kaufmann (2007)
22. Fleming, P.J., Purshouse, R.C., Lygoe, R.J.: Many-Objective Optimization: An Engineering Design Perspective. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 14–32. Springer, Heidelberg (2005)
23. Zitzler, E., Brockhoff, D., Thiele, L.: The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 862–876. Springer, Heidelberg (2007)
24. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, pp. 105–145. Springer, Berlin (2005)
25. Okabe, T., Jin, Y., Sendhoff, B.: A critical survey of performance indices for multiobjective optimisation. In: 2003 IEEE Congress on Evolutionary Computation (CEC 2003), pp. 878–885. IEEE Press, Canberra (2003)
26. Siegel, S.: Nonparametric Statistics for the Behavioral Sciences. McGraw-Hill, New York (1956)

A Hybrid Dynamic Multi-objective Immune Optimization Algorithm Using Prediction Strategy and Improved Differential Evolution Crossover Operator

Yajuan Ma, Ruochen Liu, and Ronghua Shang

Key Laboratory of Intelligent Perception and Image Understanding
of Ministry of Education of China, Xidian University, Xi'an, 710071, China

Abstract. In this paper, a hybrid dynamic multi-objective immune optimization algorithm is proposed. In the algorithm, when a change in the objective space is detected, aiming to improve the ability of responding to the environment change, a forecasting model, which is established by the non-dominated antibodies in previous optimum locations, is used to generate the initial antibodies population. Moreover, in order to speed up convergence, an improved differential evolution crossover with two selection strategies is proposed. Experimental results indicate that the proposed algorithm is promising for dynamic multi-objective optimization problems.

Keywords: Prediction Strategy, differential evolution, dynamic multi-objective, immune optimization algorithm.

1 Introduction

Many real-world systems have different characteristics in different time. Dynamic single-objective optimization has received more attention in the past [10]. Recently, people have focused on dynamic multi-objective optimization (DMO) problems [5].

In DMO problems, the objective function, constraint or the associated problem parameters may change over time, and the DMO problems often aim to trace the movement of the Pareto front (PF) and the Pareto Set (PS) within the given computation budget. If the existed classical static multi-objective techniques are applied to DMO problems directly, they will have many limitations because of lacking of the ability to react change quickly. To this end, a correct prediction of the new location of the changed PS is of great interest. Hatzakis [4] proposed a forward-looking approach to predict the new locations of the only two anchor points. Zhou [1] proposed a forecasting model to predict the new location of individuals from the location changes that have occurred in the history time environment. In this paper, we use the forecasting model [1] to guide future search. The main difference between our method and [1] is similarity detection which is used to detect whether a significant change take place in the system. Solution re-evaluation is used as similarity detection in [1], we use the population statistical information to detect environment. Moreover, if the historical information is too little to form a forecasting model, we perturb the last PS location to get the initial individuals. In the late stages of evolution, the forecasting model is used to predict the new individuals' locations.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 435–444, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Recently, applying an immune system for dynamic optimization arouse much attraction due to its natural capability of reacting to new threats. Zhang [13] suggested a dynamic multi-objective immune optimization algorithm (DMIOA) to deal with DMO problems, in which, the dimension of the design space is time-variant. Shang [9] proposed a clone selection algorithm (CSADMO) with a well-known non-uniform mutation strategy to solve DMO problems. In our paper, static multi-objective immune algorithm with non-dominated neighbor-based selection (NNIA) [8] is extended to solve DMO problems. However, NNIA may be trapped in local optimal Pareto front and converge to only a point when current non-dominated antibodies selected for proportional cloning are very few. In order to solve this problem, an improved differential evolution (DE) crossover is proposed. Different from classic DE, two selection parent individuals' strategies are used to generate new antibodies in the improved DE crossover.

2 Theoretical Background

2.1 The Definition of DMO Problems and Antibody Population

In this paper, we will solve the following DMO problems:

$$\begin{cases} \min F(x,t) = (f_1(x,t), f_2(x,t), \ldots, f_m(x,t))^T \\ s.t.\ \ x \in X \end{cases} \tag{1}$$

Where $t = 0,1,2,\ldots$ represent time. $x \subset R^D$ is the decision space and $x = (x_1, \ldots x_l) \in R^D$ is the decision variable vector. $F : (X,t) \to R^m$ consists of m real-valued objective functions which change over time $f_i(x,t)\ i = 1,2,\ldots,m$. R^m is the objective space. In this paper, an antibody $b = (b_1, b_2, \ldots, b_l)$ is the coding of variable x, denoted by $b = e(x)$, and x is called the decoding of antibody b, expressed as $x = e^{-1}(b)$. An antibody population

$$B = \{b_1, b_2, \ldots b_n\}, b_i \in R^l, 1 \le i \le n \tag{2}$$

is a set of n-dimensional antibodies, where n is the size of antibody population B.

2.2 Forecasting Model

The forecasting model [1] is introduced briefly as follows:

Assume that the recorded antibodies in the historical time environment, i.e., $Q_t, \ldots, Q_1$ can provide information for predicting the new PS locations of $t+1$. The locations of PS of $t+1$ are seen as a function of the locations $Q_t, \ldots, Q_1$:

$$Q_{t+1} = F(Q_t, \ldots, Q_1, t) \tag{3}$$

Where Q_{t+1} represents the new location of PS at $t+1$.

Suppose that $x_1,\ldots,x_t,x_i \in Q_i, i=1,\ldots,t$ are a series of antibodies which describe the movements of the PS, a generic model to predict the new antibodies locations for the $(t+1)$ -th time environment can be described as follows:

$$x_{t+1} = F(x_t, x_{t-1},\ldots,x_{t-K+1},t) \tag{4}$$

Where K denotes the number of the previous time environment that x_{t+1} is dependent on in the forecasting model. In this paper, we set $K=3$.

Here, for an antibody $x_t \in Q_t$, its parent antibody in the pervious time environment can be defined as the nearest antibody in Q_{t-1} :

$$x_{t-1} = \arg \min_{y \in Q_{t-1}} \|y - x_t\|_2 \tag{5}$$

Once a time series is constructed for each antibody in the population, we use a simple linear model to predict the new antibody:

$$x_{t+1} = F(x_t, x_{t-1}) = x_t + (x_t - x_{t-1}) \tag{6}$$

2.3 Differential Evolution

Differential Evolution (DE) algorithm [6] is a simple and effective evolutionary algorithm for optimizing problems. The mutation operator can be described as follows:

$$V_{i,t+1} = X_{r_1,t} + F*(X_{r_2,t} - X_{r_3,t}) \tag{7}$$

Where $V_{i,t+1}$ is mutant vector, $X_{r_1,t}, X_{r_2,t}, X_{r_3,t}$ are three different individuals in population, F is a mutation factor. Then combine the current vector $X_{i,t}$ and the mutant vector $V_{i,t+1}$ to form the trial vector $U_{i,t+1}$:

$$U_{i,t+1} = (U_{1,t+1}, U_{2,t+1}, \cdots, U_{N,t+1})$$
$$U_{ij,t+1} = \begin{cases} V_{ij,t+1} & if\ (rand(0,1) \le CR)\ or\ j = j_{rand} \\ X_{ij,t} & if\ (rand(0,1) > CR)\ or\ j \ne j_{rand} \end{cases} \quad i = 1,2,\cdots N,\ j = 1,2,\cdots D \tag{8}$$

Where $rand(0,1)$ is a random number within [0, 1], CR is a control parameter to determine the probability of crossover, j_{rand} is a randomly chosen index from $[1, D]$.

3 Proposed Algorithm

3.1 Similarity Detection and Prediction Mechanism

The aim of similarity detection is to detect whether a change happens, and if a change is detected, whether the adjacent time environment is similar to each other. Two methods are usually used as similarity detection. One method is solution re-evaluation

[5] [1]. A few solutions is randomly selected to evaluate them, if there is a change in any of the objectives and constraint functions, it is recognized that a change take place in the problem. In this paper, the population statistical information [7] is used as similarity detection operator. It can be formulated as follows:

$$\varepsilon(t) = \frac{\sum_{j=1}^{N_\delta} \left\| \frac{(f_j(X,t) - f_j(X,t-1))}{R(t) - U(t)} \right\|}{N_\delta} \tag{9}$$

Where, $R(t)$ is composed of the maximum value of each dimensions of $f(X,t)$ and $U(t)$ is composed of the minimum value of each dimensions of $f(X,t)$. N_δ is the size of solutions used to test the environment change. If the $\varepsilon(t)$ is greater than a predefined threshold, we think that a significant change has taken place in the system, and then the Forecasting Model is used to predict the new location of individuals. The prediction strategy is described as:

```
The prediction strategy (Output: the initial antibodies
population Q_t(0)):
Randomly select 5 sentry antibodies from Q_{t-1}(τ_T), and then
use Similarity Detection to detect environment.
If change is significant, do
    if t<3
        Q_t(0)← Perturb 20% of Q_{t-1}(τ_T) with Gauss noise
    else
        Q_t(0)←Forecasting Model
    end if
end if
```

Where $Q_{t-1}(\tau_T)$ is the optimal antibody population of the time t+1.

3.2 The Proposed Dynamic Multi-objective Immune Optimization Algorithm

The flow of the hybrid dynamic multi-objective immune optimization algorithm (HDMIO) is shown as follows:

```
The main pseudo-code of HDMIO (Output: every time
environment PS: Q_1,...,Q_{T_max}):
Initialize P(0) randomly, and get Non-dominated
population B(0), select n_A less-crowded non-dominated
solutions from B(0) to form Active Population A(0),
Set t=0;
while t<T_max do
  if t>0, do
    Conduct prediction strategy and get P_t(0), then find
    B_t(0) and A_t(0);
  end if
```

```
g = 0;
While g < τ_T, do
  C_t(g) ← Proportional clone A_t(g);
  C_t'(g) ← The improved DE crossover and polynomial
       mutation;
  C_t'(g) ∪ B_t(g) ← Combine C_t'(g) and B_t(g);
  B_t(g+1), A_t(g+1) ← C_t'(g) ∪ B_t(g)
  g = g + 1;
end while
Q_t = B_t(g);
t = t + 1
end while
```

Where g is the generation counter, τ_T is the number of generations in time environment t. T_{max} is the maximum number of time steps, n_D is maximum size of Non-dominated Population. $A_t(g)$ is Active Population, and n_A is maximum size of Active Population. $C_t(g)$ is Clone Population, and n_c is size of Clone Population.

At time t, similarity detection is applied to determining whether the new re-initialization strategy is used. After proportional clone, the improved DE crossover and polynomial mutation are operated on clone population, and then the non-dominated antibodies are identified and selected from $C_t'(g) \cup B_t(g)$. When the number of non-dominated antibodies is greater than the maximum limitation n_D and the size of non-dominated Population n_D is greater than the maximum size of Active Population n_A, both the reduction of non-dominated Population and the selection of active antibodies use the crowding-distance [3].

In the proposed algorithm, proportional cloning can be denoted as follows:

$$d_i = \left\lceil n_c \times \frac{r_i}{\sum_{i=1}^{n_A} r_i} \right\rceil \tag{10}$$

Where, r_i denotes the normalized crowding-distance value of the active antibodies a_i, $d_i, i = 1, 2, \ldots, n_A$ is the cloning number assigned to i-th active antibody, and $d_i = 1$ denotes that there is no cloning on antibody a_i.

3.3 Improved DE Crossover Operator

When selecting some antibodies to generate new antibodies in DE, a hybrid selection mechanism is used, which include selection 1 and selection 2. As Fig. 1, the antibodies in active population are less-crowded antibodies selected from non-dominated population. Proportional cloning those antibodies in Active Population and get clone population. Every time environment, in the early stages of evolution, selection 1 is used, when the current generation is larger than a pre-defined number,

the selection 2 is active. In two selection strategies, we choose the base parent $X_{r1,t}$ from the clone population randomly. While the methods of selecting other two parents $X_{r2,t}$ and $X_{r3,t}$ are different in two selection strategies. In selection 1, they are selected from non-dominated Population randomly, and in selection 2, they are randomly selected from clone population.

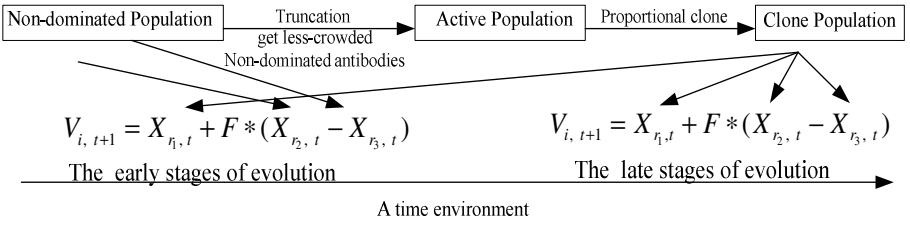


Fig. 1. Illustration of two parent antibodies selection mechanisms in DE

4 Experimental Studies

4.1 Benchmark Problems

Four different problems are tested in this paper. In DMOP1 [7] and DMOP4 [7], the optimal PS change, and the optimal PF does not change. In DMOP2 [2], the optimal PS does not change, and the optimal PF change. In DMOP3 [2], both the optimal PS and PF change. The first three problems have two objectives, and the last problem has three objectives. Fig.2 shows the true PSs and PFs of DMOPs when they are changing with time.

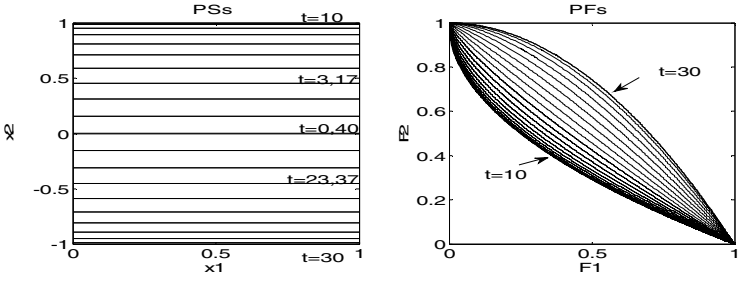


Fig. 2. Illustration of PSs and PFs of DMOps when they are changing with time

4.2 Experiments on Prediction Scheme and the Improved DE Crossover Operator

The algorithms in comparison are all conducted under the framework of dynamic NNIA. Table 1 lists six algorithms. Parameters settings are as follows: $n_D = n_c = 100$, $n_A = 20$, the severity of change $n_T = 10$, the frequency of change $\tau_T = 50$, $T_{max} = 30$, the thresh hold of $\varepsilon(t)$ is 2e-02, the parameters of DE is set

to be: $F = 0.5, CR = 0.1$ .Inverted generational distance (IGD) [12] is used for measuring the performance of the algorithms, the lower values of IGD represent good convergence ability. We used $\overline{IGD}$ to denote the average IGD value of all time environments. Fig. 3 gives the tracking of IGD in 10 time steps, and the $\overline{IGD}$ and its standard variance (std) of 20 independent runs are listed in Table 2.

Table 1. Indexs of different algorithms

Index	Algorithms
1	DNNIA-res: restart 20% non-dominated antibodies randomly
2	DNNIA-res-DE: restart scheme and DE crossover operator
3	DNNIA-gauss: perturb 20% non-dominated antibodies with Gaussian noise
4	DNNIA-gauss-DE: perturb scheme and DE crossover operator
5	DNNIA-pre: prediction scheme
6	HDMIO: prediction scheme and DE crossover operator

Taking the re-initialization scheme into consideration only, from Fig.3, we can see that, for DMOP1, DMOP3 and DMOP4, the advantage of prediction scheme is much more distinct than other re-initialization, perturb scheme is poor slightly, restart scheme works worst. When $0 < t < 3$, results of all the algorithms are very similar, since the quality of history information stored too small to form forecasting model, and the prediction scheme is in essence perturb scheme. When $t > 3$, the algorithm with the prediction scheme has best performance and can react to the variations with a faster speed. For DMOP2, the stability of HDMIO is not good, and even in some time, its performance qualities are worse than those without prediction scheme. This may be because that the true PS of DMOP2 is instant all the time, prediction scheme could break the distribution of the history PS and lose efficacy.

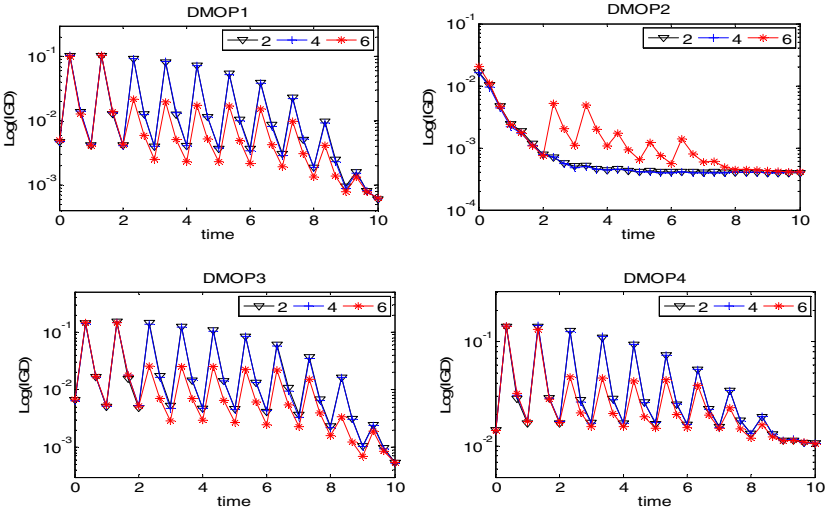


Fig. 3. *IGD* versus 10 time steps of DNNIA with different re-initialization

As the influence of new DE crossover operator to the result, from the Table 2, we can see that the *IGD* value can be improved to a certain extent for DMOP1, DMOP3 and DMOP4. For DMOP2, combining with prediction scheme, the new DE crossover operator does not improve the *IGD* value.

Table 2. Comparison of $\overline{IGD}$ of DNNIA with different re-initialization and crossover

		DNNIA-res	DNNIA-res-DE	DNNIA-gauss	DNNIA-gauss-DE	DNNIA-pre	HDMIO
DMOP1	mean	1.33E-02	3.28E-03	4.76E-03	3.24E-03	2.62E-03	**2.15E-03**
	std	3.32E-02	8.17E-05	1.83E-04	1.03E-04	1.16E-04	**7.84E-05**
DMOP2	mean	1.66E-03	1.03E-03	1.60E-03	**1.02E-03**	1.53E-03	1.74E-03
	std	1.20E-03	1.68E-04	7.21E-04	**2.45E-04**	1.20E-03	3.76E-04
DMOP3	mean	9.87E-01	3.92E-03	5.67E-01	3.83E-03	3.16E-03	**2.57E-03**
	std	2.01E+00	1.51E-04	1.40E+00	1.51E-04	1.45E-04	**7.67E-05**
DMOP4	mean	2.75E-02	1.51E-02	2.50E-02	1.51E-02	1.86E-02	**1.41E-02**
	td	1.70E-03	2.55E-04	1.60E-03	1.93E-04	1.30E-03	**1.66E-04**

4.3 Experiment of Comparing HDMIO with Other Three Different Dynamic Multi-objective Optimization Algorithms

In this section, we compared HDMIO with other four dynamic multi-objective optimization algorithms. They are DNAGAII-A [5], DNSGAII-B [5] and CSADMO [9]. For all these algorithms, $\tau_0 = 100$, the population size $N = 100$, in DNSGAII-A

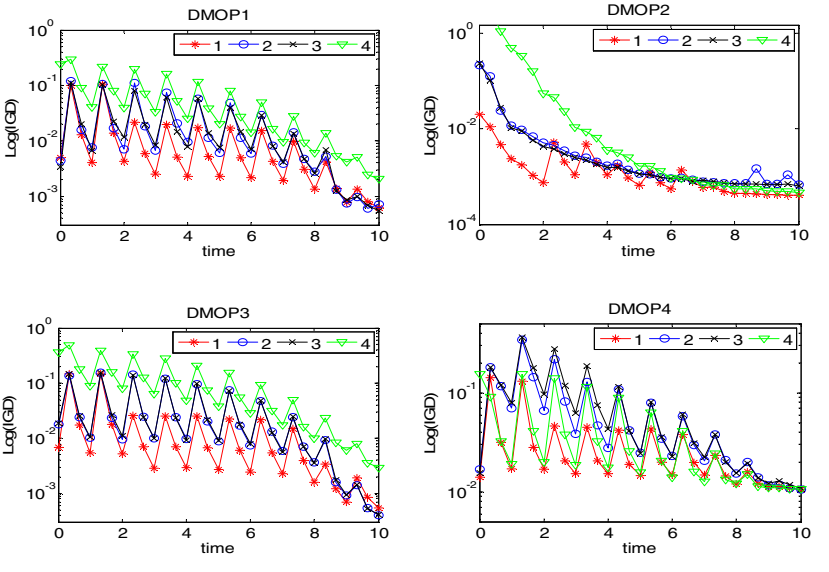


Fig. 4. *IGD* versus 10 time steps of DNNIA and other three different dynamic multi-objective optimization algorithms.1 represents HDMIO, 2 represents DNSGAII-A, 3 represents DNSGAII-B, 4 represents CSADMO

and DNNIA-B, $p_c = 1$, $p_m = 1/n$, where n is the dimension of decision variable, the parameters of HDMIO are same with the precious section. In every $t > 1$, for DNSGAII-A, DNSGAII-B and HDMIO, the number of fitness evaluations is $FEs = 5000$, while the clone proportion of CSADMO is 3, and its $FEs = 15000$. Fig. 4 shows the tracking of *IGD* in 10 time steps in details.

From Fig. 4, we can see that, for DMOP1and DMOP3, although it is difficult to form the forecasting model in first three steps, our algorithm is still superior to other three algorithms. As time goes on, the advantage of our algorithm is remarkable, and the ability to react to change is fastest. For DMOP2, the performance stability of our algorithm is poor slightly. For DMOP4, HDMIO achieve best performance in the early stages, while CSADMO works best in the late stages.

5 Conclusion

In this paper, we present a hybrid dynamic multi-objective immune optimization algorithm, in which, two mechanisms including a prediction mechanism and a new crossover operator is proposed. We use two sets of experiments to prove the effectiveness of the proposed algorithm, the first set of experiments demonstrate that the prediction mechanism can significantly improve the ability of responding to the environment, and the new crossover operator can enhance the convergence of proposed algorithm. It is concluded that the proposed algorithm for the classic DMO problems are encouraging and promising. While, when the change of the PS is insignificant or the PS is instant over time, the stability of our algorithm is not very good. So this problem is our priority for the future research.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grant (No.60803098 and No.61001202), and the Provincial Natural Science Foundation of Shaanxi of China (No. 2010JM8030 and No. 2009JQ8015).

References

1. Zhou, A.M., Jin, Y.C., Zhang, Q.F., Sendhoff, B., Tsang, E.: Prediction-Based Population Re-Initialization for Evolutionary Dynamic Multi-Objective Optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 832–846. Springer, Heidelberg (2007)
2. Goh, C.K., Tan, K.C.: A competitive –cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Transactions on Evolutionary Computation 13(1), 103–127 (2009)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
4. Hatzakis, I., Wallace, D.: Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach. In: Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2006), Seattle, Washington, USA, pp. 1201-1208 (2006)

5. Deb, K., Bhaskara, U.N., Karthik, S.: Dynamic Multi-objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydro-thermal Power Scheduling. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 803–817. Springer, Heidelberg (2007)
6. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution. A Practical Approach to Global Optimization. Springer, Berlin (2005) ISBN 3-540-29859-6
7. Farina, M., Amato, P., Deb, K.: Dynamic multi-objective optimization problems: Test cases, approximations and applications. IEEE Transactions on Evolutionary Computation 8(5), 425–442 (2004)
8. Gong, M.G., Jiao, L.C., Du, H.F., Bo, L.F.: Multi-objective immune algorithm with nondominated neighbor-based selection. Evolutionary Computation 16(2), 225–255 (2008)
9. Shang, R., Jiao, L., Gong, M., Lu, B.: Clonal Selection Algorithm for Dynamic Multiobjective Optimization. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-m., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005, Part I. LNCS (LNAI), vol. 3801, pp. 846–851. Springer, Heidelberg (2005)
10. Yang, S.X., Yao, X.: Population-Based Incremental Learning With Associative Memory for Dynamic Environments. IEEE Transactions on Evolutionary Computation 12(5), 542–561 (2008)
11. Zhang, Z.H., Qian, S.Q.: Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control. Applied Soft Computing 8, 959–971 (2008)
12. Van Veldhuizen, D.A.: Multi-Objective evolutionary algorithms: Classification, analyzes, and new innovations (Ph.D. Thesis). Wright-Patterson AFB: Air Force Institute of Technology (1999)

Optimizing Interval Multi-objective Problems Using IEAs with Preference Direction

Jing Sun[1,2], Dunwei Gong[1], and Xiaoyan Sun[1]

[1] School of Information and Electrical Engineering,
China University of Mining and Technology, Xuzhou, China
[2] School of Sciences,
Huai Hai Institute of Technology, Lianyungang, China

Abstract. Interval multi-objective optimization problems (MOPs) are popular and important in real-world applications. We present a novel interactive evolutionary algorithm (IEA) incorporating an optimization-cum-decision-making procedure to obtain the most preferred solution that fits a decision-maker (DM)'s preferences. Our method is applied to two interval MOPs and compared with PPIMOEA and the *posteriori* method, and the experimental results confirm the superiorities of our method.

Keywords: Evolutionary algorithm, Interaction, Multi-objective optimization, Interval, Preference direction.

1 Introduction

When handling optimization problems in real-world applications, it is usually necessary to simultaneously consider several conflicting objectives. Furthermore, due to many objective and/or subjective factors, these objectives and/or constraints frequently contain uncertain parameters, e.g., fuzzy numbers, random variables, and intervals. These problems are called uncertain MOPs. For many practical problems, compared with creating the precise probability distributions of random variables or the member function of fuzzy numbers, the bounds of the uncertain parameters can be much more easily identified [1]. We focus on MOPs with interval parameters [2] in this study. The mathematical model of this problem can be formulated as follows:

$$\begin{aligned} &\max f(\mathbf{x},\mathbf{c}) = (f_1(\mathbf{x},\mathbf{c}), f_2(\mathbf{x},\mathbf{c}), \cdots, f_m(\mathbf{x},\mathbf{c}))^T \\ &\text{s.t.}\mathbf{x} \in \mathbf{S} \subseteq \mathbf{R}^n \\ &\mathbf{c} = (c_1, c_2, \cdots, c_l)^T, c_k = [\underline{c}_k, \overline{c}_k], k = 1, 2, \cdots, l \end{aligned} \tag{1}$$

where $\mathbf{x}$ is an n-dimensional decision variable, $\mathbf{S}$ is a decision space of $\mathbf{x}$, $f_i(\mathbf{x},\mathbf{c})$ is the i-th objective function with interval parameters for each $i = 1, 2, \cdots, m$, $\mathbf{c}$ is an interval vector parameter, where c_k is the k-th component of $\mathbf{c}$ with $\underline{c}_k$ and $\overline{c}_k$ being its lower and upper limits, respectively. Each objective value in problem (1) is an interval due to its interval parameters, and the i-th objective value is denoted as $f_i(\mathbf{x},\mathbf{c}) \triangleq [\underline{f}_i(\mathbf{x},\mathbf{c}), \overline{f}_i(\mathbf{x},\mathbf{c})]$.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 445–452, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Evolutionary algorithms (EAs) are a kind of globally stochastic optimization methods inspired by nature evolution and heredity mechanisms. Since EAs can simultaneously search for several Pareto optimal solutions in one run, they become efficient methods, such as NSGA-II [3], of solving MOPs. EAs for MOPs with interval parameters [2] aim to find a set of well-converged and evenly-distributed Pareto optimal solutions. However, in practice, it is necessary to arrive at the DM's most preferred solution [4]. The methods can be grouped into the following three categories, i.e., *a priori* methods, *a posteriori* methods, and interactive methods.

There have been many interactive evolutionary multi-objective optimization methods for MOPs with deterministic parameters [4]-[7], however, there exists few interactive method for MOPs with interval parameters. To our best knowledge, there only exists our recently proposed method, named solving interval MOPs using EAs with preference polyhedron (PPIMOEA) [8].

Types of preference information asked from the DM include reference points [5], reference directions [6], and so on. For interactive based reference points/ directions methods, reference points and directions are expressed as the form of aspiration levels, which are comfortable and intuitive for the DM [5]. In the initial stage of evolution, the DM has no overview of the objective space and his/her aspiration levels are blind.

The DM's preference information can be acquired by pairwise comparing all optimal solutions, which can be used to construct his/her preference model. For preference cone based methods [7], it is necessary to select the best and the worst ones from the objective values corresponding to the alternatives. Compared with the method of specifying aspiration levels, it is much easier to select the worst value, which alleviates the cognitive burden on the DM. A preference polyhedron of [8] indicates the DM's preference region and points out his/her preference direction.

Given the above ideas, we propose an IEA for interval MOP based on preference direction by employing the framework of NSGA-II, which incorporates an optimization-cum-decision-making procedure. This algorithm makes the best of the DM's preference information, and a preference direction is elicited from the preference polyhedron. In addition, an interval achievement scalarizing function is constructed by taking the worst value and the preference direction as the reference point and direction, respectively. The above function is used to rank optimal solutions and direct the search to the DM's preference region.

The remaining of this paper is organized as follows: Section 2 expounds framework of our algorithm. The applications of our method in typical bi-objective optimization problems with interval parameters are given in section 3. Section 4 outlines the main conclusions of our work and suggests possible opportunities to be further researched.

2 Proposed Algorithm

We propose an IEA for MOPs with interval parameters based on the preference polyhedron in this section. Having evolved τ generations by an EA for MOPs

with interval parameters, the DM is provided with $\eta \geq 2$ optimal solutions with large crowding/approximation metrics from the non-dominated solutions every τ generations, and chooses the worst one from the objective values corresponding to them. With these optimal solutions sent to the DM, a preference polyhedron is created in the objective space, and his/her preference direction is elicited from it, expounded in subsection 2.1. Till the next τ generations, the constructed preference polyhedron and an approximation metric, described in subsection 2.2, based on the above direction are used to modify the domination principle, elaborated in subsection 2.3. When the termination criterion is met, the first superior individual in the population is the DM's most preferred solution.

2.1 Preference Direction

For the theory of the preference polyhedron, please refer to [8]. From the theorems, the gray region in Fig. 1 is the DM's preferred one, which implicitly shows the DM's preference direction, and the rest is either the DM's non-preferred or uncertain preference one. If the population evolves along the preference direction, the algorithm will rapidly find the DM's most preferred solution. To this end, we need to elicit the preference direction from the preference polyhedron. For the sake of simplicity, we choose the middle direction of the preference polyhedron as the DM's preference direction. The detailed method of eliciting the preference direction from the preference polyhedron in the two-dimensional case is as follows. The discussion is divided into the following two cases:

(1) When a component of the worst value is the minimal value of corresponding objective, the directions of direction vectors of the two lines are selected as the ones whose direction cosine in the objective, i.e. the component in the objective, are larger than 0;

(2) When a component of the worst value is not the minimal value of corresponding objective, if the line lies above the worst value, the directions of direction vectors are selected as the ones whose direction cosine in the second objective is larger than 0; otherwise, those in the first objective are chosen.

The unit direction vectors of the two lines are denoted as $v_1 = (v_1^1, v_1^2)$ and $v_2 = (v_2^1, v_2^2)$, respectively, then the direction of the sum of the two direction vectors is the preference direction. The direction, shown as the one of $v_1 + v_2$ in Fig. 1, is the DM's preference direction.

2.2 Approximation Metric

The value of an achievement scalarizing function reflects the approximation of the objective value corresponding to an alternative to the DM's most preferred value on the Pareto front. In maximization problems, the larger the value of the achievement function, the closer the alternative to the DM's most preferred solution is.

The objective values considered here are intervals, the above real-valued achievement function is, thus, not applicable. It is necessary to replace the

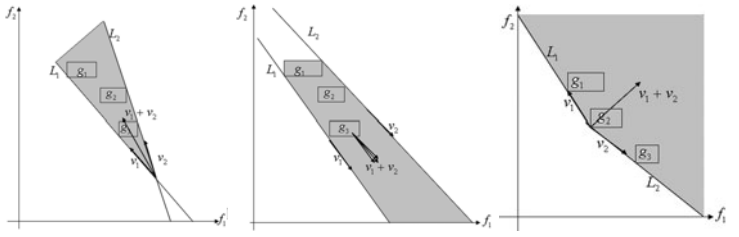


Fig. 1. Elicitation of preference direction

real-valued variables of the achievement function with interval ones. Accordingly, the following interval achievement function is got.

$$\begin{aligned} s(f(\mathbf{x},\mathbf{c}), f(\mathbf{x}_k,\mathbf{c}), r) = & \max_{i=1,\cdots,m} \left[\frac{|f_i(\mathbf{x},\mathbf{c}) - f_i(\mathbf{x}_k,\mathbf{c})|}{r^i} \right] \\ & + \rho \sum_{i=1}^{m} |f_i(\mathbf{x},\mathbf{c}) - f_i(\mathbf{x}_k,\mathbf{c})| \end{aligned} \tag{2}$$

where $f(\mathbf{x},\mathbf{c})$ is the objective value corresponding to individual $\mathbf{x}$ in the t-th generation, $f(\mathbf{x}_k,\mathbf{c})$ is the worst value, $r = (r^1, r^2, \cdots, r^m)$ is the preference direction, $|f_i(\mathbf{x},\mathbf{c}) - f_i(\mathbf{x}_k,\mathbf{c})|$ denotes the distance between $f_i(\mathbf{x},\mathbf{c})$ and $f_i(\mathbf{x}_k,\mathbf{c})$, whose definition is the maximum of $\left|\underline{f}_i(\mathbf{x},\mathbf{c}) - \underline{f}_i(\mathbf{x}_k,\mathbf{c})\right|$ and $\left|\overline{f}_i(\mathbf{x},\mathbf{c}) - \overline{f}_i(\mathbf{x}_k,\mathbf{c})\right|$ [9], where $\underline{f}_i(\mathbf{x},\mathbf{c}), \overline{f}_i(\mathbf{x},\mathbf{c})$ and $\underline{f}_i(\mathbf{x}_k,\mathbf{c}), \overline{f}_i(\mathbf{x}_k,\mathbf{c})$ are the lower and the upper limits of intervals $f_i(\mathbf{x},\mathbf{c})$ and $f_i(\mathbf{x}_k,\mathbf{c})$, respectively. ρ is a sufficiently small positive scalar. The value of this function is called the approximation metric of individual $\mathbf{x}$ in this study.

2.3 Sorting Optimal Solutions

We use the following strategy to sort the individuals: first, the dominance relation based on intervals [2] is used; then, the individuals with the same rank are classified into three categories, i.e. the preferred, the uncertain preference and the non-preferred individuals [8]; finally, the individuals with both the same rank and category are further ranked based on the approximation metric. The larger the approximation metric, the better the performance of the individual is.

The above sorting strategy is suitable to select individuals in Step 4, too.

3 Applications

The proposed algorithm's performances are confirmed by optimizing two benchmark bi-objective optimization problems and comparing it with PPIMOEA and an *a posteriori* method. The implementation environment is as follows: Pentium(R) Dual-Core CPU, 2G RAM, and Matlab7.0.1. Each algorithm is run for 20 times independently, and the averages of these results are calculated. Two bi-objective optimization problems with interval parameters, i.e. ZDT$_I$1 and ZDT$_I$4, from [2] are chosen as benchmark problems.

3.1 Preference Function

In our experiments, for ZDT_I1 and ZDT_I4, the following quasi-concave increasing value function

$$V_1(f_1, f_2) = (f_1 + 0.4)^2 + (f_2 + 5.5)^2 \tag{3}$$

and linear value function

$$V_2(f_1, f_2) = 1.25f_1 + 1.50f_2 \tag{4}$$

are used to emulate the DM to make decisions, respectively.

3.2 Parameter Settings

Our algorithm is run for 200 generations with the population size of 40. Simulated binary crossover (SBX) operator and polynomial mutation [4] are employed, and the crossover and mutation probabilities are set to 0.9 and 1/30, respectively. In addition, the distribution indexes for crossover and mutation operators with $\eta_c = 20$ and $\eta_m = 20$ are adopted, respectively. The number of decision variables, in the range of $[0, 1]$, is 30 for these two test problems. The number of individuals provided to the DM for evaluation is 3.

3.3 Performance Measures

(1) The best value of the preference function (V metric, for short). This index measures the DM's satisfaction with the optimal solution. The larger the value of V metric, the more satisfactory the DM with the optimal solution is.

(2) CPU time (T metric, for short). The smaller the CPU time of an algorithm, the higher its efficiency is.

3.4 Results and Analysis

Our experiments are divided into two groups. The first one investigates the influence of different values of τ on the performance of our algorithm. We also compare the proposed method with the *posteriori* one, i.e., the value of τ is 200, and the decision-making is executed at the end of the algorithm. The second one compares the difference between our algorithm and PPIMOEA.

Influence of τ on Our Algorithm's Performance. Fig. 2 shows the curves of V metrics of two optimization problems w.r.t. the number of generations when the value of τ is 10, 40 and 200, respectively. It can be observed from Fig. 2 that:

(1) For the same value of τ, the value of V metric increases along with the evolution of a population, indicating that the obtained solution is more and more suitable to the DM's preferences.

(2) For the same generation, the value of V metric increases along with the decrease of the value of τ, or equivalently, the increase of the interaction frequency,

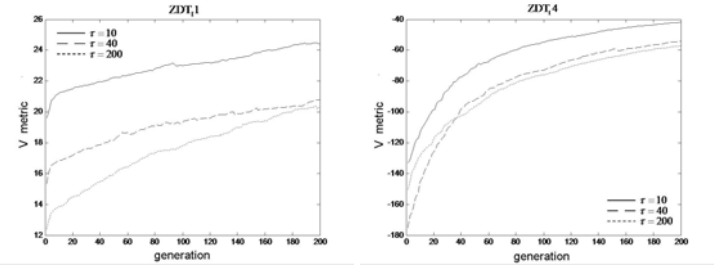


Fig. 2. Curves of V metrics w.r.t. number of generations

suggesting that the more frequent the interaction, the better the most preferred solution is. The interactive method thus obviously outperforms the *posteriori* method.

Table 1 lists the T metrics of two optimization problems for different values of τ. It can be observed from Table 1 that the value of T metric decreases along with the increase of the interaction frequency. This means that the increase of the interaction frequency can guide the search to the DM's most preferred solution quickly.

Table 1. Influence of τ on T metric (Unit: s)

τ	10	40	200
ZDT$_I$1	12.77	13.03	16.45
ZDT$_I$4	10.22	10.41	15.64

Table 2. Comparison between our method and *a posteriori* method

	a posteriori method	our method	P(0)
ZDT$_I$1 V metric	20.24	26.45	1.3e-004
T metric	16.45	13.33	7.6e-004
ZDT$_I$4 V metric	-57.10	-36.96	0.0039
T metric	15.64	10.22	9.80e-011

Table 2 shows the data of our method when $\tau = 10$ and the *posteriori* method on two performance measures. The last column gives the results of the hypotheses test, denoted as P(0). One-tailed test is utilized, and null hypothesis is that both medians are equal. It can be observed from Table 2 that our method outperforms the *posteriori* method at the significant level of 0.05.

Comparison between Our Method and PPIMOEA. The value of τ is set to be 10 in this group of experiments. Fig. 3 illustrates the values of V metrics of different methods w.r.t. the number of generations. As it can be observed from Fig. 3, for the same generation, the value of V metric of our method is larger

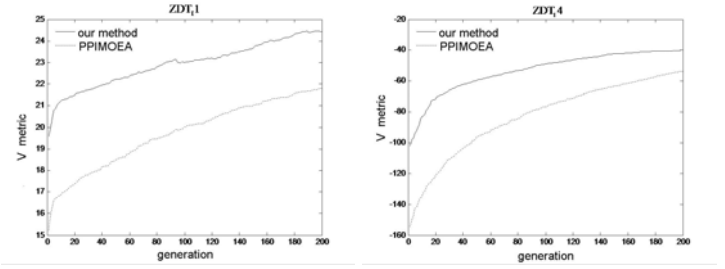


Fig. 3. V metrics of different methods w.r.t. the number of generations

that the one of PPIMOEA, indicating that the most preferred solution obtained by our method is more suitable to the DM's preferences.

Table 3 lists the data of our method and PPIMOEA on two performance measures. It can be observed from Table 3 that our algorithm outperforms PPIMOEA at the significant level of 0.05, suggesting that our method can reach the most preferred solution that more fits the DM's preferences in a short time.

Table 3. Comparison between our method and PPIMOEA

	PPIMOEA	our method	P(0)
ZDT$_I$1 V metric	21.81	26.45	0.0030
T metric	16.39	13.33	2.2e-004
ZDT$_I$4 V metric	-53.76	-36.96	0.0039
T metric	13.85	10.22	5.7e-017

4 Conclusions

MOPs with interval parameters are popular and important, few effective method of solving them, however, exists as a result of their complexity.

We focus on these problems and present an IEA for MOPs with interval parameters based on the preference direction. The DM's preference direction is elicited from a preference polyhedron, and the preference polyhedron and direction are used to rank optimal solutions. The DM's most preferred solution is finally found.

The DM's preference direction points out the search direction. If the DM's preference information is incorporated into genetic operators, e.g., crossover and mutation operators, the search performance of the algorithm will be further improved. This is our future research topic.

Acknowledgments. This work was jointly supported by National Natural Science Foundation of China, grant No. 60775044, Program for New Century Excellent Talents in Universities, grant No. NCET-07-0802, and Natural Science Foundation of HHIT, grant No. 2010150037.

References

1. Zhao, Z.H., Han, X., Jiang, C., Zhou, X.X.: A Nonlinear Interval-based Optimization Method with Local-densifying Approximation Technique. Struct. Multidisc. Optim. 42, 559–573 (2010)
2. Limbourg, P., Aponte, D.E.S.: An Optimizaiton Algorithm for Imprecise Multi-objective Problem Function. In: Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 459–466. IEEE Press, New York (2005)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6, 182–197 (2002)
4. Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.): Multiobjective Optimization - Interactive and Evolutionary Approaches. LNCS, vol. 5252. Springer, Heidelberg (2008)
5. Luque, M., Miettinen, K., Eskelinen, P., Ruiz, F.: Incorporating Preference Information in Interactive Reference Point. Omega 37, 450–462 (2009)
6. Deb, K., Kumar, A.: Interactive Evolutionary Multi-objective Optimization and Decision-making Using Reference Direction Method. Technical report, KanGAL (2007)
7. Fowler, J.W., Gel, E.S., Koksalan, M.M., Korhonen, P., Marquis, J.L., Wallenius, J.: Interactive Evolutionary Multi-objective Optimization for Quasi-concave Preference Functions. Eur. J. Oper. Res. 206, 417–425 (2010)
8. Sun, J., Gong, D.W., Sun, X.Y.: Solving Interval Multi-objective Optimization Problems Using Evolutionary Algorithms with Preference Polyhedron. In: Genetic and Evolutionary Computation Conference, pp. 729–736. ACM, NewYork (2011)
9. Moore, R.E., Kearfott, R.B., Cloud, M.J.: Introduction to Interval Analysis. SIAM, Philadelphia (2009)

Fitness Landscape-Based Parameter Tuning Method for Evolutionary Algorithms for Computing Unique Input Output Sequences

Jinlong Li[1], Guanzhou Lu[2], and Xin Yao[2]

[1] Nature Inspired Computation and Applications Laboratory (NICAL), Joint USTC-Birmingham Research Institute in Intelligent Computation and Its Applications, School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230026, China, University of Science and Technology of China, China

[2] CERCIA, School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK, University of Birmingham, UK

Abstract. Unique Input Output (UIO) sequences are used in conformance testing of Finite state machines (FSMs). Evolutionary algorithms (EAs) have recently been employed to search UIOs. However, the problem of tuning evolutionary algorithm parameters remains unsolved. In this paper, a number of features of fitness landscapes were computed to characterize the UIO instance, and a set of EA parameter settings were labeled with either 'good' or 'bad' for each UIO instance, and then a predictor mapping features of a UIO instance to 'good' EA parameter settings is trained. For a given UIO instance, we use this predictor to find good EA parameter settings, and the experimental results have shown that the correct rate of predicting 'good' EA parameters was greater than 93%. Although the experimental study in this paper was carried out on the UIO problem, the paper actually addresses a very important issue, i.e., a systematic and principled method of tuning parameters for search algorithms. This is the first time that a systematic and principled framework has been proposed in Search-Based Software Engineering for parameter tuning, by using machine learning techniques to learn good parameter values.

1 Introduction

Finite state machines (FSMs) have been usually used to model software, communication protocols and circuitslee94a. To test a state machine, state verification should be implemented. While unique input output sequence (UIO) is the most used method to tackle with state verification. In software engineering domain, search based software engineering attempts to use optimization techniques, such as Evolutionary Algorithms (EAs), for many computationally hard problems, and UIO problem was tested by [7,6].

Whether a given state has a UIO or not is an NP-hard problem pointed out by Lee and Yannakakis[5]. Guo and Derderian[7,4] have reformulated UIO problem

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 453–460, 2011.
© Springer-Verlag Berlin Heidelberg 2011

as an optimisation problem and solved it with EAs. Their experimental results have shown that EAs outperform random search on a larger FSM. Furthermore, Lehre and Yao confirmed theoretically that the expected running time of (1+1) EA on some FSM instances is polynomial, while random search needs exponential time[10].

We will focus on tackling the problem of producing UIOs with EAs. Lehre and Yao have proposed[12] three types of UIO instances: EA-easy instances, EA-hard instances and tunable difficulty instances. In addition to these instances, there are many other UIO instances that are difficult to analyze theoretically. Lehre and Yao have pointed that[11,3] crossover and non-uniform mutation are useful for some UIO instances, which means different parameters settings may seriously affect performance of solving UIO instances with EA. In this paper, we aim to develop an automated approach to set up EA parameters for effectively solving the problem of generating UIOs.

Tuning EA parameters for a given problem instance is hard. Previous work revealed that 90% of the time is spent on fine-tuning algorithm parameter settings[1]. Most of those approaches are attempting to find one parameters setting for all problem instances or an instance class[2,15,9]. The features used by those approaches are problem based and the feature selection relies on the knowledge of domain experts. For example, SATzilla[17] uses 48 features mostly specify to SAT to construct per-instance algorithm portfolios for SAT. A problem-independent feature represented by a behavior sequence of a local search procedure is used to perform instance-based automatic parameter tuning[13]. A feature of problem instance called fitness-probability cloud characterizing the evolvability of fitness landscape was proposed[14] and this feature does not require any problem knowledge to calculate and predict the performances of EAs.

In this paper, a number of fitness-probability clouds are used to characterize a problem instance, since we believe that the more features of instance we know about, the more effective algorithm for this instance will be designed. The major contributions of this paper include the following.

- We propose a number of fitness-probability clouds to characterize UIO problem instances, not just using one fitness-probability cloud to characterize one fitness landscape. To characterize an UIO instance, the knowledge of domain experts is not required, which means our method will be easily extended to other software engineering problems.
- A framework of adaptively selecting EA parameters settings is designed. We have tested our framework on the UIO problem, and the experimental results have shown that a new UIO instance will get 'good' EA parameters settings with probability greater than 93%.

2 Preliminaries

2.1 Problem Definition

Definition 1. *(Finite State Machine). A finite state machine(FSM) is a quintuple:* $\boldsymbol{M} = (\boldsymbol{S}, \boldsymbol{X}, \boldsymbol{Y}, \delta, \lambda)$*, where* $\boldsymbol{X}, \boldsymbol{Y}$ *and* $\boldsymbol{S}$ *are finite and nonempty sets of*

input symbols, output symbols, and states, respectively; $\delta : \boldsymbol{S} \times \boldsymbol{X} \longrightarrow \boldsymbol{S}$ *is the state transition function; and* $\lambda : \boldsymbol{S} \times \boldsymbol{X} \longrightarrow \boldsymbol{Y}$ *is the output function.*

Definition 2. *(Unique Input Output Sequence). An unique input output sequence for a given state* s_i *is an input/output sequence* x/y*, where* $x \in \boldsymbol{X}^*, y \in \boldsymbol{Y}^*, \forall s_j \neq s_i, \lambda(s_i, x) \neq \lambda(s_j, x)$ *and* $\lambda(s_i, x) = y$*.*

There maybe exist $k(\geq 0)$ UIOs for a given state. Suppose x/y is a UIO for state $s \in \mathbf{S}$, *concatenation*(x, x') will produce another UIO's input string for state s, where $\forall x' \in \mathbf{S}$, which means we can deduce infinitely many UIOs for state s.

To compute UIOs with EAs, in this paper candidate solutions are represented by input strings restricted to $\mathbf{X}^n = \{0, 1\}^n$, where n is the number of states of FSM. In general, the length of shortest UIO is unknown, and so we assume that our objective is to search for a UIO of input string length n for state s_1 in all FSM instances.

The fitness function is defined as a function of the state partition tree[7,10,11].

Definition 3. *(UIO fitness function[10,11]). For a FSM* M *with* n *states, the fitness function* $f : \boldsymbol{X}^n \longrightarrow \mathbb{N}$ *is defined as* $f(x) := n - \gamma_M(s, x)$*, where* s *is the initial state for which we want to find a UIO, and* $\gamma_M(s, x) := |\{t \in \boldsymbol{S} | \lambda(s, x) = \lambda(t, x)\}|$*.*

There are $|\mathbf{X}|^n$ candidate solutions with $n - 1$ different values. A candidate solution x^* is a global optimum if and only if x^* produces a UIO and $f(x^*) = n - 1$.

2.2 Evolutionary Algorithm and Its Parameters

Here, we solve UIO problem with evolutionary algorithms (EAs) usually called target algorithms. The detailed steps of EA are shown as *Algorithm 1.*

Algorithm 1. $(\mu + \lambda)$- Evolutionary Algorithms

Choose μ initial solutions $\mathbf{P}^{(0)} = \{x_1^{(0)}, x_2^{(0)}, \ldots, x_\mu^{(0)}\}$ uniformly at random from $\{0, 1\}^n$
$k \longleftarrow 0$
while *termination criterion is no met* **do**
  $\mathbf{P}_m^{(k)} \longleftarrow \mathcal{N}_j(\mathbf{P}^{(k)})$ %%mutation operator
  $\mathbf{P}^{(k+1)} \longleftarrow \mathcal{S}_i(\mathbf{P}^{(k)}, \mathbf{P}_m^{(k)})$ %%selection operator
  $k \longleftarrow k + 1$
end

In this paper, $(\mu + \lambda) - EAs$ described by *Algorithm 1* have three kinds of parameters: population sizes, neighborhood operators, selection operators.

- Population sizes: We provide 3 different $(\mu+\lambda)$ options: $\{(4+4), (7+3), (3+7)\}$.

– Neighborhood operators $\mathcal{N}_j, (j = 1, 2, \ldots, 12)$: There are 3 types of neighborhood operators with different mutation probabilities.
  - $\mathcal{N}_1(x) \sim \mathcal{N}_5(x)$: Bit-wised mutation, flip each bit with probability $p = c/n$, where $c \in \{0.5, 1, 2, n/2, n-1\}$, and n is problem size;
  - $\mathcal{N}_6(x) \sim \mathcal{N}_9(x)$: c bits flip, uniformly at random select c bits to flip, where $c = \{1, 2, n/2, n-1\}$;
  - $\mathcal{N}_{10}(x) \sim \mathcal{N}_{12}(x)$: Non-uniform mutation[3], for each bit $i, 1 \leq i \leq n$, flip it with probability $\chi(i) = c/(i+1)$, where $c = \{0.5, 1, 2\}$.

  These 12 neighborhood operators will be used to act on UIO fitness function, and then generate 12 fitness-probability clouds to characterize a UIO instance.
– Selection operators $\mathcal{S}_i, (i = 1, 2)$: Two selection schemes will be considered in this paper.
  - Truncation Selection: Sort all individuals in $\mathbf{P}^{(k)}$ and $\mathbf{P}_m^{(k)}$ by their fitness values, then select μ best individuals as the next generation $\mathbf{P}^{(k+1)}$.
  - Roulette Wheel Selection: Retain all the best individuals in $\mathbf{P}^{(k)}$ and $\mathbf{P}_m^{(k)}$ directly, and the rest of the individuals of population are selected by roulette wheel.

For a given UIO instance, there are 72 different EA parameter combinations which can be looked as 72 different EA parameters settings, and our goal is to find 'good' settings for a given UIO instance. In *Algorithm 1*, the terminating criterion is satisfied when a UIO has been found.

3 Fitness-Probability Cloud

In the parameters tuning framework proposed, Fitness-Probability Clouds (fpc) have been employed as characterisations of the problem instance. fpc is initially proposed in [14] and is briefly reviewed here.

3.1 Escape Probability

The notion of Escape Probability (Escape Rate) is introduced by Merz [16] to quantify a factor that influences the problem hardness for EAs. In theoretical runtime analysis of EAs, He and Yao [8] proposed an analytic way to estimate the mean first hitting time of an absorbing Markov chain, in which the transition probability between states were used. To make the study of Escape Probability applicable in practice, we adopt the idea of transition probability in a Markov chain. Let us partition the search space into $L+1$ sets according to fitness values, $F = \{f_0, f_1, \ldots, f_L \quad | \quad f_0 < f_1 < \cdots < f_L\}$ denotes all possible fitness values of the entire search space. S_i denotes the average number of steps required to find an improving move starting in an individual of fitness values f_i. The escape probability $P(f_i)$ is defined as $P(f_i) = \dfrac{1}{S_i}$.

The greater the escape probability for a particular fitness value f_i, the easier it is to improve the fitness quality.

3.2 Fitness-Probability Cloud

We can extend the definition of escape probability to be on a set of fitness values. P_i denotes the average escape probability for individuals of fitness value equal to or above f_i and is defined as: $P_i = \frac{\sum_{f_j \in C_i} P(f_j)}{|C_i|}$, where $C_i = \{f_j | j \geq i\}$.

If we take into account all the P_i for a given problem, this would be a good indication of the degree of evolvability of the problem. For this reason, the Fitness-Probability Cloud (fpc) is defined as: $fpc = \{(f_0, P_0), \ldots, (f_L, P_L)\}$.

3.3 Accumulated Escape Probability

It is clear by definition that the Fitness-Probability Cloud (fpc) can demonstrate certain properties related to evolvability and problem hardness, however, the mere observation is not sufficient to quantify these properties. Hence we define a numerical measure called Accumulated Escape Probability (aep) based on the concept of fpc: $aep = \frac{\sum_{f_i \in F} P_i}{|F|}$, where $F = \{f_0, f_1, ..., f_L \quad | \quad f_0 < f_1 < ... < f_L\}$.

4 Adaptive Selection of EA Parameters

This framework consists of two phases. The first phase is mainly for training the predictor based on the existing data sets, then the features of new problem instances would be fed into the predictor and produce 'good' parameters settings in the second phase.

4.1 The First Phase: Training Predictor

We are using Support Vector Machines (SVM) to train the predictor.

First, training data structure is denoted by a tuple $\mathbf{D} = (\mathbf{F}, \mathbf{PC}, \mathbf{L})$. $\mathbf{F}$ represents the features of the problem instance. For a UIO instance, it is a vector of fitness-probability clouds[14]. Fitness-probability cloud is a useful feature to characterise the fitness landscape. One neighbourhood operator produces one distinct fitness-probability cloud; the more neighbourhood operators acting on fitness function, the more features of fitness function will be generated. This paper adopts 12 common neighbourhood operators in the literature to generate 12 fitness-probability clouds for characterizing UIO instance.

$\mathbf{PC}$ of tuple $\mathbf{D}$ is the ID of an EA parameters settings. Each problem instance represented by its features $\mathbf{F}$ is solved by target algorithm with 72 parameters settings, and the performances are evaluated by the number of fitness evaluations, $\mathbf{E}_{ij}$ denotes the performance of the target algorithm with parameters setting j on problem instance i is $\mathbf{E}_{ij}$, where $j = 1, 2, \ldots, 72$.

$\mathbf{L}$ represents the categorical feature of the training data. The value of 'good' or 'bad' was labelled according to fitness evaluations of target algorithm with

parameters setting **PC**. A parameters setting is 'good' if the fitness evaluations of target algorithm EA with given setting is less than a threshold v.

To generate training data, m problem instances are randomly selected, denoted by $\mathbf{P} = \{p_1, p_2, \ldots, p_m\}$. For each problem instance, a set of neighbourhood operators $\mathcal{N}_i, i = 1, 2, \ldots, 12$ are applied to generate the corresponding Accumulated Escape Probability (aep) as its features. We end up with a vector $(aep_1, aep_2, \ldots, aep_{12})$ as the features of the problem instance. The categorical features will then labelled after executing EAs with different parameters settings on the problem instances.

The data sets used are identified to possess characteristics like small samples and imbalance data sets. In light of the characteristics of the data sets, given Support Vector Machine is a popular machine learning algorithm which can handle small samples, we employ a support vector machine classifier.

4.2 The Second Phase: Predicting 'good' EA Parameters

Once the predictor is trained, for a new UIO instance, we can calculate its features $(aep_1, aep_2, \ldots, aep_{12})$ and then use them as input to the predictor to find good EA parameters settings.

5 Experimental Studies

In order to test our framework, 24 UIO instances have been generated at random, the problem size across all instances is 20. We applied the approach described in Section 3 to generate the training data. The stopping criteria is set to 'found the optima'. EAs with each parameter setting is executed for 100 times on each UIO instance. For each UIO instance, 72 different settings produce 72 different samples, thus we have 1728 samples including training data and testing data partitioned randomly, and 10×10-fold cross validation will be adopted to evaluate our method.

We are interested in 'good' EA Parameters Settings (gEAPC), and the best EA parameters setting having the smallest fitness evaluations on an instance was labeled 'good' in our experiments, the remaining 71 settings were labeled 'good' or 'bad' depending on the differences between their fitness evaluations and the threshold value v. We let $v = pr \times \bar{\mathbf{E}}_i$, where $\bar{\mathbf{E}}_i$ is the mean value of fitness evaluations on instance i and pr replaces v to regulate the number of gEAPC.

As shown in Table 1, the number of gEAPC (2nd column '#gEAPC') in all 1728 samples was decreasing while we were reducing pr. For one UIO instance must at least have one gEAPC in practice, and the ideal result is that the predictor gives just one best EA parameters setting, but when we set pr larger, too many gEAPC will be labeled and almost half of all settings are 'good' which means predicting results useless for us to select gEAPC. The smaller the value of pr, the less gEAPC we will have, but the correct rate of predicting gEAPC, denoted by s_g in Table 1, is decreasing when pr is smaller than 0.1. Furthermore, we found out that more and more instances have no gEAPC predicted when

Table 1. Correct rates of predicting gEAPC with different values of pr. Values of s_g in 3rd column, the average of 10×10 fold cross validation, are equal to $(Correctly\ Classified\ gEAPC/Total\ Number\ of\ gEAPC)$.

pr	#gEAPC	s_g	gEAPC found?	pr	#gEAPC	s_g	gEAPC found?
0.7	1180	0.500	yes	0.6	1115	0.510	yes
0.5	1007	0.709	yes	0.45	955	0.690	yes
0.4	874	0.689	yes	0.35	806	0.685	yes
0.3	716	0.726	yes	0.25	604	0.689	yes
0.2	489	0.653	yes	0.18	441	0.656	yes
0.16	391	0.709	yes	0.15	377	0.694	yes
0.14	343	0.698	yes	0.135	328	0.632	yes
0.13	326	0.687	yes	0.125	306	0.875	yes
0.12	299	0.764	yes	0.115	286	0.903	yes
0.11	267	0.933	yes	0.1	237	0.925	no
0.09	200	0.861	no	0.08	177	0.864	no
0.05	71	0.782	no	0.01	50	0.620	no

decreasing value of pr, and 4th column of Table 1 will be 'no' if there exists any testing instance without predicted gEAPC. Table 1 shows that the best value of pr was 0.11 and there are about 267 gEAPC and all instances will have at least one predicted gEAPC.

6 Conclusions

EA parameter setting significantly affects the performance of the algorithm. This paper presents a learning-based framework to automatically select 'good' EA parameter settings. The UIO problem has been used to evaluate this framework, experimental results showed that by properly setting the values of v or pr, the framework can learn at least one good parameter setting for each problem instance tested.

Future work includes testing our framework on a wider range of problems and investigating the influence of the machine learning techniques employed, via studies on techniques other than the Support Vector Machine.

Acknowledgments. This work was partially supported by an EPSRC grant (No. EP/D052785/1) and NSFC grants (Nos. U0835002 and 61028009). Part of the work was done while the first author was visiting CERCIA, School of Computer Science, University of Birmingham, UK.

References

1. Adenso-Diaz, B., Laguna, M.: Fine-tuning of algorithms using fractional experimental design and local search. Operations Research 54(1), 99–114 (2006)
2. Birattari, M., Stuzle, T., Paquete, L., Varrentrapp, K.: A racing algorithm for configuring metaheuristics. In: Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation, GECCO 2002, pp. 11–18. Morgan Kaufmann (2002)

3. Cathabard, S., Lehre, P.K., Yao, X.: Non-uniform mutation rates for problems with unknown solution lengths. In: Proceedings of the 11th Workshop Proceedings on Foundations of Genetic Algorithms, FOGA 2011, pp. 173–180. ACM, New York (2011)
4. Derderian, K., Hierons, R.M., Harman, M., Guo, Q.: Automated unique input output sequence generation for conformance testing of fsms. The Computer Journal 49 (2006)
5. Lee, D., Yannakakis, M.: Testing finite-state machines: state identification and verification. IEEE Transactions on computers 43(3), 30–320 (1994)
6. Guo, Q., Hierons, R., Harman, M., Derderian, K.: Constructing multiple unique input/output sequences using metaheuristic optimisation techniques. IET Software 152(3), 127–140 (2005)
7. Guo, Q., Hierons, R.M., Harman, M., Derderian, K.: Computing Unique Input/Output Sequences Using Genetic Algorithms. In: Petrenko, A., Ulrich, A. (eds.) FATES 2003. LNCS, vol. 2931, pp. 164–177. Springer, Heidelberg (2004)
8. He, J., Yao, X.: Towards an analytic framework for analysing the computation time of evolutionary algorithms. Artificial Intelligence 145, 59–97 (2003)
9. Hutter, F., Hoos, H.H., Leyton-brown, K., Sttzle, T.: Paramils: An automatic algorithm configuration framework. Journal of Artificial Intelligence Research 36, 267–306 (2009)
10. Lehre, P.K., Yao, X.: Runtime analysis of (1+l) ea on computing unique input output sequences. In: IEEE Congress on Evolutionary Computation, 2007, pp. 1882–1889 (September 2007)
11. Lehre, P.K., Yao, X.: Crossover can be Constructive When Computing Unique Input Output Sequences. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abbass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 595–604. Springer, Heidelberg (2008)
12. Lehre, P.K., Yao, X.: Runtime analysis of the (1+1) ea on computing unique input output sequences. Information Sciences (2010) (in press)
13. Lindawati, Lau, H.C., Lo, D.: Instance-based parameter tuning via search trajectory similarity clustering (2011)
14. Lu, G., Li, J., Yao, X.: Fitness-probability cloud and a measure of problem hardness for evolutionary algorithms. In: Proceedings of the 11th European Conference on Evolutionary Computation in Combinatorial Optimization, EvoCOP 2011, pp. 108–117. Springer, Heidelberg (2011)
15. Maturana, J., Lardeux, F., Saubion, F.: Autonomous operator management for evolutionary algorithms. Journal of Heuristics 16, 881–909 (2010)
16. Merz, P.: Advanced fitness landscape analysis and the performance of memetic algorithms. Evol. Comput. 12, 303–325 (2004)
17. Xu, L., Hutter, F., Hoos, H., Leyton-Brown, K.: Satzilla: Portfolio-based algorithm selection for sat. Journal of Artificial Intelligence Research 32, 565–606 (2008)

Introducing the Mallows Model on Estimation of Distribution Algorithms

Josu Ceberio, Alexander Mendiburu, and Jose A. Lozano

Intelligent Systems Group
Faculty of Computer Science
The University of The Basque Country
Manuel de Lardizabal pasealekua, 1
20018 Donostia - San Sebastian, Spain
jceberio001@ikasle.ehu.es,
{alexander.mendiburu,ja.lozano}@ehu.es
http://www.sc.ehu.es/isg

Abstract. Estimation of Distribution Algorithms are a set of algorithms that belong to the field of Evolutionary Computation. Characterized by the use of probabilistic models to learn the (in)dependencies between the variables of the optimization problem, these algorithms have been applied to a wide set of academic and real-world optimization problems, achieving competitive results in most scenarios. However, they have not been extensively developed for permutation-based problems. In this paper we introduce a new EDA approach specifically designed to deal with permutation-based problems. In this paper, our proposal estimates a probability distribution over permutations by means of a distance-based exponential model called the Mallows model. In order to analyze the performance of the Mallows model in EDAs, we carry out some experiments over the Permutation Flowshop Scheduling Problem (PFSP), and compare the results with those obtained by two state-of-the-art EDAs for permutation-based problems.

Keywords: Estimation of Distribution Algorithms, Probabilistic Models, Mallows Model, Permutations, Flow Shop Scheduling Problem.

1 Introduction

Estimation of Distribution Algorithms (EDAs) [10, 15, 16] are a set of Evolutionary Algorithms (EAs). However, unlike other EAs, at each step of the evolution, EDAs learn a probabilistic model from a population of solutions trying to explicitly express the interrelations between the variables of the problem. The new offspring is then obtained by sampling the probabilistic model. The algorithm stops when a certain criterion is met, such as a maximum number of generations, homogeneous population, or lack of improvement in the last generations.

Many different approaches have been given in the literature to deal with permutation problems by means of EDAs. However, most of these proposals

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 461–470, 2011.
© Springer-Verlag Berlin Heidelberg 2011

are adaptations of classical EDAs designed to solve discrete or continuous domain problems. Discrete domain EDAs follow the path-representation codification [17] to encode permutation problems. These approaches learn, departing from a dataset of permutations, a probability distribution over a set $\Omega = \{0, \ldots, n-1\}^n$, where $n \in \mathbb{N}$. Therefore, the sampling of these models has to be modified in order to provide permutation individuals. Algorithms such as Univariate Marginal Distribution Algorithm (UMDA), Estimation of Bayesian Networks Algorithm (EBNAs), or Mutual Information Maximization for Input Clustering (MIMIC) have been applied with this encoding to different problems [2, 11, 17].

Adaptations of continuous EDAs [3, 11, 17] use the Random Keys representation [1] to encode a solution with random numbers. These numbers are used as sort keys to obtain the permutation. Thus, to encode a permutation of length n, each index in the permutation is assigned a value (key) from some real domain, which is usually taken to be the interval [0, 1]. Subsequently, the indexes are sorted using the keys to get the permutation. The main advantage of random keys is that they always provide feasible solutions, since each encoding represents a permutation. However, solutions are not processed in the permutation space, but in the largely redundant real-valued space. For example, for length 3 permutation, strings (0.2, 0.1, 0.7) and (0.4, 0.3, 0.5) represent the same permutation (2, 1, 3).

The limitations of these direct approaches, both in the discrete and continuous domains, encouraged the research community of EDAs to implement specific algorithms for solving permutation-based problems. Bosman and Thierens introduced the ICE [3, 4] algorithm to overcome the bad performance of Random Keys in permutation optimization. The ICE replaces the sampling step with a special crossover operator which is guided by the probabilistic model, guaranteeing feasible solutions.

In [18] a new framework for EDAs called Recursive EDAs (REDAs) is introduced. REDAs is an optimization strategy that consists of separately optimizing different subsets of variables of the individual.

Tsutsui et al. [19, 20] propose two new models to deal with permutation problems. The first approach is called Edge Histogram Based Sampling Algorithm (EHBSA). EHBSA builds an Edge Histogram Matrix (EHM), which models the edge distribution of the indexes in the selected individuals. A second approach called Node Histogram Based Sampling Algorithm (NHBSA), introduced later by the authors, models the frequency of the indexes at each absolute position in the selected individuals. Both algorithms simulate new individuals by sampling the marginals matrix. In addition, the authors proposed the use of a template-based method to create new solutions. This method consists of randomly choosing an individual from the previous generation, dividing it into c random segments and sampling the indexes for one of the segments, leaving the remaining indexes unchanged. A generalization of these approaches was given by Ceberio et al. [6], where the proposed algorithm learns k-order marginal models.

As stated in [5], Tsutsui's EHBSA and NHBSA approaches yield the best results for several permutation-based problems, such as Traveling Salesman Problem, Flow Shop Scheduling Problem, Quadratic Assignment Problem or Linear Ordering Problem. However, these approaches are still far from achieving optimal solutions, which means that there is still room for improvement. Note that the introduced approaches do not estimate a probability distribution over the space of permutations that allow us to calculate the probability of a given solution in a closed form. Motivated by this issue and working in that direction, we present a new EDA which models an explicit probability distribution over the permutation space: the Mallows EDA.

The remainder of the paper is as follows: Section 2 introduces the optimization problem we tackle: The Permutation Flow Shop Scheduling Problem. In Section 3 the Mallows model is introduced. In section 4, some preliminary experiments are run to study the behavior of the Mallows EDA. Finally, conclusions are drawn in Section 5.

2 The Permutation Flowshop Scheduling Problem

The Flowshop Scheduling Problem [9] consists of scheduling n jobs ($i = 1, \ldots, n$) with known processing time on m machines ($j = 1, \ldots, m$). A job consists of m operations and the j-th operation of each job must be processed on machine j for a specific time. A job can start on the j-th machine when its $(j-1)$-th operation has finished on machine $(j-1)$, and machine j is free. If the jobs are processed in the same order on different machines, the problem is named as Permutation Flowshop Scheduling Problem (PFSP). The objective of the PFSP is to find a permutation that achieves a specific criterion such as minimizing the total flow time, the makespan, etc. The solutions (permutations) are denoted as $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ where σ_i represents the job to be processed in the i-th position. For instance, in a problem of 4 jobs and 3 machines, the solution $(2, 3, 1, 4)$, indicates that job 2 is processed first, next job 3 and so on.

Let $p_{i,j}$ denote the processing time for job i on machine j, and $c_{i,j}$ denote the completion time of job i on machine j. Then, $c_{\sigma_i,j}$ is the completion time of the job scheduled in the i-th position on machine j. $c_{\sigma_i,j}$ is computed as $c_{\sigma_i,j} = p_{\sigma_i,j} + max\{c_{\sigma_i,j-1}, c_{\sigma_{i-1},j}\}$. As this paper addresses the makespan performance measure, the objective function F is defined as follows:

$$F(\sigma_1, \sigma_2, \ldots, \sigma_n) = c_{\sigma_n,m}$$

As can be seen, the solution of the problem is given by the processing time of the last job σ_n in the permutation, since this is the last job to finish.

3 The Mallows Model

The Mallows model [12] is a distance-based exponential probability model over permutation spaces. Given a distance d over permutations, it can be defined

by two parameters: the central permutation σ_0, and the spread parameter θ. (1) shows the explicit form of the probability distribution over the space of permutations:

$$P(\sigma) = \frac{1}{\psi(\theta)} e^{-\theta d(\sigma, \sigma_0)} \tag{1}$$

where $\psi(\theta)$ is a normalization constant. When $\theta > 0$, the central permutation σ_0 is the one with the highest probability value and the probability of the other $n! - 1$ permutations exponentially decreases with the distance to the central permutation (and the spread parameter θ). Because of these properties, the Mallows distribution is considered analogous to the Gaussian distribution on the space of permutations (see Fig. 1). Note that when θ increases, the curve of the probability distribution becomes more peaked at σ_0.

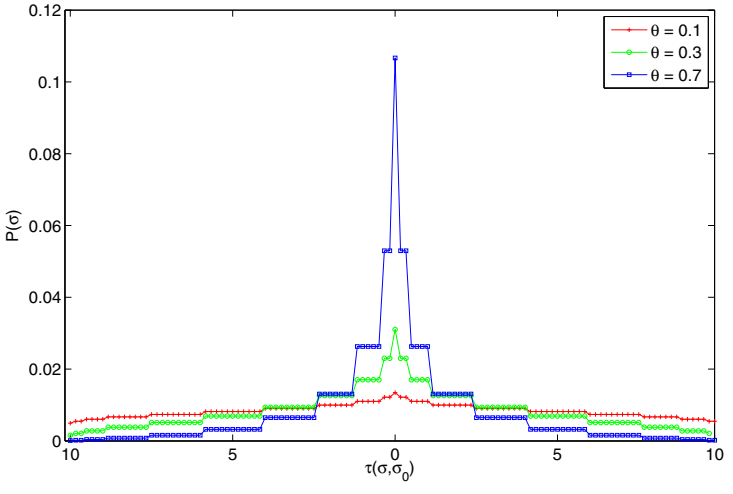


Fig. 1. Mallows probability distribution with the Kendall-τ distance for different spread parameters. In this case, the dimension of the problem is $n = 5$.

3.1 Kendall-τ Distance

The Mallows model is not tied to a specific distance. In fact, it has been used with different distances in the literature such as Kendall, Cayley or Spearman [8]. For the application of the Mallows model in EDAs, we have chosen the Kendall-τ distance. This is the most commonly used distance with the Mallows model, and in addition, its definition resembles the structure of a basic neighborhood system in the space of permutations. Given two permutations σ_1 and σ_2, the Kendall-τ distance counts the total number of pairwise disagreements between both of them i.e., the minimum number of adjacent swaps to convert σ_1 into σ_2. Formally, it can be written as

$$\tau(\sigma_1, \sigma_2) = |\{(i,j) : i < j, (\sigma_1(i) < \sigma_1(j) \wedge \sigma_2(i) > \sigma_2(j)) \\ \vee (\sigma_2(i) < \sigma_2(j) \wedge \sigma_1(i) > \sigma_1(j))\}|.$$

The above metric can be equivalently written as

$$\tau(\sigma_1, \sigma_2) = \sum_{j=1}^{n-1} V_j(\sigma_1, \sigma_2) \tag{2}$$

where $V_j(\sigma_1, \sigma_2)$ is the minimum number of adjacent swaps to set in the j-th position of σ_1, $\sigma_1(j)$, the value $\sigma_2(j)$. This decomposition allows to factorize the distribution as a product of independent univariate exponential models[14], one for each V_j and that (see (3) and (4)).

$$\psi(\theta) = \prod_{j=1}^{n-1} \psi_j(\theta) = \prod_{j=1}^{n-1} \frac{1 - e^{-(n-j+1)\theta}}{1 - e^{-\theta}} \tag{3}$$

$$P(\sigma) = \frac{e^{-\theta \sum_{j=1}^{n-1} V_j(\sigma, \sigma_0)}}{\prod_{j=1}^{n-1} \psi_j(\theta)} = \prod_{j=1}^{n-1} \frac{e^{-\theta V_j(\sigma, \sigma_0)}}{\psi_j(\theta)} \tag{4}$$

This property of the model is essential to carry out an efficient sampling. Furthermore, one can uniquely determine any σ by the $n-1$ integers $V_1(\sigma)$, $V_2(\sigma)$,..., $V_{n-1}(\sigma)$ defined as

$$V_j(\sigma, I) = \sum_{l>j} 1_{[l \prec_\sigma j]} \tag{5}$$

where I denotes the identity permutation (1, 2,... n) and $l \prec_\sigma j$ means that l precedes j (i.e. is preferred to j) in permutation σ.

3.2 Learning and Sampling a Mallows Model

At each step of the EDA, we need to learn a Mallows model from the set of selected individuals (permutations). Therefore, given a dataset of permutations $\{\sigma_0, \sigma_1, \ldots, \sigma_N\}$ we need to estimate σ_0 and θ. In order to do that, we use the maximum likelihood estimation method. The log-likelihood function can be written as

$$log\ l(\sigma_1, ..., \sigma_N | \sigma_0, \theta) = -N \sum_{j=1}^{n-1} (\theta \bar{V}_j + \log \psi_j(\theta)) \tag{6}$$

where $\bar{V}_j = \sum_{i=1}^{N} V_j(\sigma_i, \sigma_0)/N$, i.e. $\bar{V}_j$ denotes the observed mean for V_j.

The problem of finding the central permutation or consensus ranking is called rank aggregation and it is, in fact, equivalent to finding the MLE estimator of σ_0, which is NP-hard. One can find several methods for solving this problem, both exact [7] and heuristic [13]. In this paper we propose the following: first, the

average of the values at each position is calculated, and then, we assign index 1 to the position with the lowest average value, next index 2 to the second lowest position, and so on until all the n values are assigned.

Once σ_0 is known, the estimation of θ maximizing the log-likelihood is immediate by numerically solving the following equation:

$$\sum_{j=1}^{n-1} \bar{V}_j = \frac{n-1}{e^{\theta}-1} - \sum_{j=1}^{n-1} \frac{n-j+1}{e^{(n-j+1)\theta}-1} \tag{7}$$

In general, this solution has no closed form expression, but can be solved numerically by standard iterative algorithms such as Netwon-Rapshon.

In order to sample, we consider a bijection between the V_j-s and the permutations. By sampling the probability distribution of the V_j-s defined by (8), a V_j-s vector is obtained. The new permutations are calculated applying the sampled V_j vector to the consensus permutation σ_0 following a specific algorithm [14].

$$P[V_j(\sigma\sigma_0^{-1}, I) = r] = \frac{e^{-\theta r}}{\psi_j(\theta)} \tag{8}$$

4 Experiments

Once the Mallows model has been introduced, we devote this section to carrying out some experiments in order to analyze the behavior of this new EDA. As stated previously, the variance of the Mallows model is controlled by a spread parameter θ, and therefore it will be necessary to observe how the model behaves according to different values of θ. In a second phase, and based on the values previously obtained, the Mallows EDA will be run for some instances of the FSP problem. In addition, for comparison purposes, two state-of-the-art EDAs [5] will be also included, in particular Tsutsui's EHBSA and NHBSA approaches.

4.1 Analysis of the Spread Parameter θ

As can be seen in the description of the Mallows model, the spread parameter θ will be the key to control the trade-off between exploration and exploitation. As shown in Fig. 1, as the value of θ increases, the probability tends to concentrate on a particular permutation (solution). In order to better analyze this behavior, we have run some experiments, varying the values of θ and observing the probability assigned to the consensus ranking (σ_0). Instances of different sizes (10, 20, 50, and 100) and a wide range of θ values (from 0 to 10) have been studied. The results shown in Fig. 2 demonstrate how, for low values of θ, the probability of σ_0 is quite small, thus encouraging a exploration stage. However, once a threshold is exceeded, the probability assigned to σ_0 increases quickly, leading the algorithm to an exploitation phase.

Based on these results, we completed a second set of experiments executing the Mallows EDA on some FSP instances. The θ parameter was fixed using a

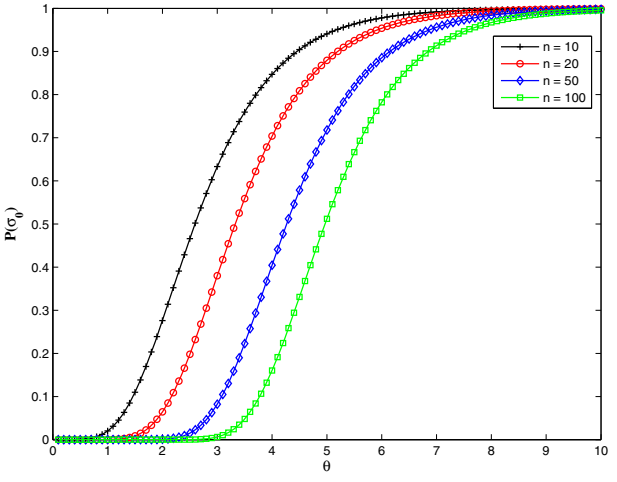


Fig. 2. Probability assigned to σ_0 for different θ and n values

range of promising values extracted from the previous experiment. Particularly, we decided to use 8 values in the range [0,2]. These values are {0.00001, 0.0001, 0.001, 0.01, 0.1, 0.5, 1, 2}. The rest of the parameters typically used in EDAs are presented in Table 1. Regarding the FSP instances, the first instance of each set *tai20×5, tai20×10, tai50×10*, *tai100×10* and *tai100×20*[1] was selected. Each experiment was run 10 times. Table 2 shows the error rate of these executions. This error rate is calculated as the normalized difference between the best value obtained by the algorithm and the best known solution.

Table 1. Execution parameters of the algorithms. Being n the problem size.

Parameter	Value
Population size	$10n$
Selection size	$10n/2$
Offspring size	$10n - 1$
Selection type	Ranking selection method
Elitism selection method	The best individual of the previous generation is guaranteed to survive
Stopping criteria	$100n$ maximum generations or $10n$ maximum generations without improvement

The results shown in 2 indicate that the lowest or highest values of θ (in the [0,2] interval) provide the worst results, and as θ moves inside the interval the performance increases. Particularly, the best results are obtained for 0.1, 0.5 and 1 values.

[1] Éric Taillard's web page. http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html

Table 2. Average error rate of the Mallows EDA with different constant θs

θ	20×5	20×10	50×10	100×10	100×20
0.00001	0.0296	0.0930	0.1359	0.0941	0.1772
0.0001	0.0316	0.0887	0.1342	0.0917	0.1748
0.001	0.0295	0.0982	0.1369	0.0910	0.1765
0.01	0.0297	0.0954	0.1275	0.0776	0.1629
0.1	0.0152	0.0694	0.0847	**0.0353**	**0.1142**
0.5	**0.0081**	0.0347	**0.0780**	0.0408	0.1236
1	0.0125	**0.0333**	0.0936	0.0610	0.1444
2	0.0182	0.0601	0.1192	0.0781	0.1649

4.2 Testing the Mallows EDA on FSP

Finally, we decided to run some preliminary tests for the Mallows EDA algorithm on the previously introduced set of FSP instances (taking in this case the first six instances from each file). Taking into account the results extracted from the analysis of θ, we decided to fix its initial value to 0.001, and to set the upper bound to 1. The parameters described in Table 1 were used for the EDAs. In particular, for NHBSA and EHBSA algorithms, B_{ratio} was set to 0.0002 as suggested by the author in [20].

For each algorithm and problem instance, 10 runs have been completed. In order to analyze the effect of the population size on the Mallows model, in addition to $10n$ we have also tested n, $5n$ and $20n$ sizes.

Table 3 shows the average error and standard deviation of the Mallows EDA and Tsutsui's approaches regarding the best known solutions. Note that each entry in the table is the average of 60 values (6 instances × 10 runs). Looking at these results, it can be seen that Tsutsui's approaches yield better results for small instances. However, as the size of the problem grows, both approaches obtain similar results for 50 × 20 instances, and the Mallows EDA shows a better performance for the biggest instances 100 × 10 and 100 × 20. The results obtained show that the Mallows EDA is better for almost all population sizes. These results stress the potential of this Mallows EDA approach for permutation-based problems.

Table 3. Average error and standard deviation for each type of problem. Results in bold indicate the best average result found.

EDA		Mallows				EHBSA	NHBSA
		n	$5n$	$10n$	$20n$	$10n$	$10n$
20×5	*avg.*	0.0137	0.0102	0.0102	0.0096	**0.0039**	0.0066
	dev.	0.0042	0.0037	0.0035	0.0039	0.0034	0.0032
20×10	*avg.*	0.0357	0.0258	0.0250	0.0232	**0.0065**	0.0076
	dev.	0.0054	0.0033	0.0037	0.0030	0.0023	0.0016
50×10	*avg.*	0.0392	0.0345	0.0342	0.0349	**0.0323**	0.033
	dev.	0.0067	0.0071	0.0059	0.0062	0.0066	0.0069
100×10	*avg.*	0.0093	**0.0078**	0.0083	0.0089	0.0199	0.0157
	dev.	0.0040	0.0040	0.0045	0.0053	0.0047	0.0062
100×20	*avg.*	**0.0583**	0.0610	0.0661	0.0587	0.0676	0.0631
	dev.	0.0116	0.0130	0.0132	0.0121	0.0050	0.0071

5 Conclusions and Future Work

In this paper a specific EDA for dealing with permutation-based problems was presented. We introduced a novel EDA, that unlike previously designed permutation based EDAs, is intended for codifying probabilities over permutations by means of the Mallows model. In order to analyze the behavior of this new proposal, several experiments have been conducted. Firstly, the θ parameter has been analyzed, in an attempt to discover its influence in the exploration-exploitation trade-off. Secondly, the Mallows EDA has been executed over several FSP instances using the information extracted from θ values in the initial experiments. Finally, for comparison purposes, two state-of-the-art EDAs have been executed: EHBSA and NHBSA. From these preliminary results, it can be concluded that the Mallows EDA approach presents an interesting behavior, obtaining better results than Tsutsui's algorithms as the size of the problem increases.

As future work, there are several points that deserve a deeper analysis. On the one hand, it would be interesting to extend the analysis of θ in order to obtain a better understanding of its influence: initial value, upper bound, etc. On the other hand, with the aim of ratifying these initial results it would be interesting to test this Mallows EDA on a wider set of problems, such as the Traveling Salesman Problem, the Quadratic Assignment Problem or the Linear Ordering Problem.

Acknowledgments. We gratefully acknowledge the generous assistance and support of Ekhine Irurozki and Prof. S. Tsutsui in this work. This work has been partially supported by the Saiotek and Research Groups 2007-2012 (IT-242-07) programs (Basque Government), TIN2010-14931 and Consolider Ingenio 2010 - CSD 2007 - 00018 projects (Spanish Ministry of Science and Innovation) and COMBIOMED network in computational biomedicine (Carlos III Health Institute). Josu Ceberio holds a grant from Basque Goverment.

References

1. Bean, J.C.: Genetic Algorithms and Random Keys for Sequencing and Optimization. INFORMS Journal on Computing 6(2), 154–160 (1994)
2. Bengoetxea, E., Larrañaga, P., Bloch, I., Perchant, A., Boeres, C.: Inexact graph matching by means of estimation of distribution algorithms. Pattern Recognition 35(12), 2867–2880 (2002)
3. Bosman, P.A.N., Thierens, D.: Crossing the road to efficient IDEAs for permutation problems. In: Spector, L., et al. (eds.) Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2001, pp. 219–226. Morgan Kaufmann, San Francisco (2001)
4. Bosman, P.A.N., Thierens, D.: Permutation Optimization by Iterated Estimation of Random Keys Marginal Product Factorizations. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 331–340. Springer, Heidelberg (2002)

5. Ceberio, J., Irurozki, E., Mendiburu, A., Lozano, J.A.: A review on Estimation of Distribution Algorithms in Permutation-based Combinatorial Optimization Problems. Progress in Artificial Intelligence (2011)
6. Ceberio, J., Mendiburu, A., Lozano, J.A.: A Preliminary Study on EDAs for Permutation Problems Based on Marginal-based Models. In: Krasnogor, N., Lanzi, P.L. (eds.) GECCO, pp. 609–616. ACM (2011)
7. Cohen, W.W., Schapire, R.E., Singer, Y.: Learning to order things. In: Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems, NIPS 1997, vol. 10, pp. 451–457. MIT Press, Cambridge (1998)
8. Fligner, M.A., Verducci, J.S.: Distance based ranking Models. Journal of the Royal Statistical Society 48(3), 359–369 (1986)
9. Gupta, J., Stafford, J.E.: Flow shop scheduling research after five decades. European Journal of Operational Research (169), 699–711 (2006)
10. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Dordrecht (2002)
11. Lozano, J.A., Mendiburu, A.: Solving job schedulling with Estimation of Distribution Algorithms. In: Larrañaga, P., Lozano, J.A. (eds.) Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation, pp. 231–242. Kluwer Academic Publishers (2002)
12. Mallows, C.L.: Non-null ranking models. Biometrika 44(1-2), 114–130 (1957)
13. Mandhani, B., Meila, M.: Tractable search for learning exponential models of rankings. In: Artificial Intelligence and Statistics (AISTATS) (April 2009)
14. Meila, M., Phadnis, K., Patterson, A., Bilmes, J.: Consensus ranking under the exponential model. In: 22nd Conference on Uncertainty in Artificial Intelligence (UAI 2007), Vancouver, British Columbia (July 2007)
15. Mühlenbein, H., Paaß, G.: From Recombination of Genes to the Estimation of Distributions I. Binary Parameters. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996, Part IV. LNCS, vol. 1141, pp. 178–187. Springer, Heidelberg (1996)
16. Pelikan, M., Goldberg, D.E.: Genetic Algorithms, Clustering, and the Breaking of Symmetry. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, Springer, Heidelberg (2000)
17. Robles, V., de Miguel, P., Larrañaga, P.: Solving the Traveling Salesman Problem with EDAs. In: Larrañaga, P., Lozano, J.A. (eds.) Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers (2002)
18. Romero, T., Larrañaga, P.: Triangulation of Bayesian networks with recursive Estimation of Distribution Algorithms. Int. J. Approx. Reasoning 50(3), 472–484 (2009)
19. Tsutsui, S.: Probabilistic Model-Building Genetic Algorithms in Permutation Representation Domain Using Edge Histogram. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 224–233. Springer, Heidelberg (2002)
20. Tsutsui, S., Pelikan, M., Goldberg, D.E.: Node Histogram vs. Edge Histogram: A Comparison of PMBGAs in Permutation Domains. Technical report, Medal (2006)

Support Vector Machines with Weighted Regularization

Tatsuya Yokota and Yukihiko Yamashita

Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152–8550, Japan
yokota@yy.ide.titech.ac.jp, yamasita@ide.titech.ac.jp
http://www.titech.ac.jp

Abstract. In this paper, we propose a novel regularization criterion for robust classifiers. The criterion can produce many types of regularization terms by selecting an appropriate weighting function. L2 regularization terms, which are used for support vector machines (SVMs), can be produced with this criterion when the norm of patterns is normalized. In this regard, we propose two novel regularization terms based on the new criterion for a variety of applications. Furthermore, we propose new classifiers by applying these regularization terms to conventional SVMs. Finally, we conduct an experiment to demonstrate the advantages of these novel classifiers.

Keywords: Regularization, Support vector machine, Robust classification.

1 Introduction

In this paper, we discuss binary classification methods based on a discriminant model. Essentially, linear models, which consist of basis functions and their parameters, are often used as discriminant models. In particular, kernel classifiers, which are types of linear models, play an important role in pattern classification, such as classification based on support vector machines (SVMs) and kernel Fisher discriminants (KFDs) [3, 4, 6, 12].

In general, a criterion for learning is based on minimization of the regularization term and the cost function. There exist various cost functions, such as squared loss, hinge loss, logistic loss, L1-loss, and Huber's robust loss [2, 5, 9, 10]. On the other hand, there is only a small variety of regularization terms(where L2 norm or L1 norm is usually used [11]) because it is considered meaningless to treat the parameters unequally for the regression problem.

In this paper, we propose a novel regularization criterion for robust classifiers. The criterion is given as a positive weighting function and a discriminant model, and its regularization term takes the form of a convex quadratic term. The criterion is considered an extension of one with an L2 norm since the proposed term can produce regularization with an L2 norm. Also, we propose two regularization terms by choosing the weighting functions according to the distribution of

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 471–480, 2011.
© Springer-Verlag Berlin Heidelberg 2011

patterns. Novel classifiers can be created by replacing these regularization terms with basic regularization terms (i.e., L2-norm terms) in SVMs. This classification procedure, which includes not only new classifiers but also basic SVMs, is referred to as "weighted regularization SVM" (WR-SVM).

If we assign a large weight in a regularization term to a certain area where differently labeled patterns are mixed or outliers are included, the classifier should become robust. Thus, we propose the use of two types of weighting functions. One function is the Gaussian distribution function, which can be used to strongly regularize areas where differently labeled patterns are mixed. The other function, which is based on the difference of two Gaussian distributions, can be used to strongly regularize areas including outliers. In fact, it is necessary to perform high-order integrations to obtain the proposed regularization terms. However, we can obtain these regularization terms analytically by using the above-mentioned weighting functions and the Gaussian kernel model. We denote these classifiers as "$\mathrm{SVM_G}$" and "$\mathrm{SVM_D}$", respectively.

The rest of this paper is organized as follows. In Section 2, general classification criteria are explained. In Section 3, we describe the proposed regularization criterion and the new regularization terms, and also prove that this criterion includes the L2 norm. In Section 4, we present the results of experiments conducted in order to analyze the properties of $\mathrm{SVM_G}$ and $\mathrm{SVM_D}$. In Section 5, we discuss the proposed approach and classifiers in depth. Finally, we provide our conclusions in Section 6.

2 Criterion for Classification

In this section, we recall classification criteria based on discriminant functions.

Let $y \in \{+1, -1\}$ be the category to be estimated from a pattern $\boldsymbol{x}$. We have independent and identically distributed samples: $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$. A discriminant function is denoted by $D(\boldsymbol{x})$, and estimated category $\hat{y}$ is given by $\hat{y} = \mathrm{sign}[D(\boldsymbol{x})]$. We define the basic linear discriminant function as

$$D(\boldsymbol{x}) := \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b, \tag{1}$$

where

$$\boldsymbol{w} = \begin{pmatrix} w_1 & w_2 & \cdots & w_M \end{pmatrix}^T, \quad \boldsymbol{x} = \begin{pmatrix} x_1 & x_2 & \cdots & x_M \end{pmatrix}^T \tag{2}$$

are respectively a parameter vector and a pattern vector, and b is a bias parameter. Although this is a rather simple model, if we replace the pattern vector $\boldsymbol{x}$ with a function $\boldsymbol{\phi}(\boldsymbol{x})$ with an arbitrary basis, we notice that this model includes the kernel model and all other linear models. We discuss such models later in this paper.

Most classification criteria are based on minimization of regularization and loss terms. If we let $R(D(\boldsymbol{x}), y)$ and $L(D(\boldsymbol{x}), y)$ be respectively a regularization term and a loss function, the criterion is given as

$$\text{minimize} \quad R(D(\boldsymbol{x}), y) + c \sum_{n=1}^{N} L(D(\boldsymbol{x}_n), y_n), \tag{3}$$

where c is an adjusting parameter between two terms. We often use

$$R := ||\boldsymbol{w}||^2 \tag{4}$$

as L2 regularization. This is a highly typical regularization term, and it is used in most classification and regression methods. Combining (4) with the hinge loss function, we obtain the criterion for support vector machines (SVMs) [12]. Furthermore, regularization term (4) and the squared loss function provide the regularized least squares regression (LSR). In this way, a wide variety of classification and regression methods can be produced by choosing a combination of a regularization term and a loss function.

3 Weighted Regularization

In this section, we define a novel criterion for regularization and explain its properties.

Let a weighting function $Q(\boldsymbol{x})$ satisfy

$$Q(\boldsymbol{z}) > 0 \tag{5}$$

for all $\boldsymbol{z} \in \mathcal{D}$, where $\mathcal{D}$ is our data domain. The new regularization criterion is given by

$$R := \int_{\mathcal{D}} Q(\boldsymbol{x}) |\langle \boldsymbol{w}, \boldsymbol{x} \rangle|^2 \mathrm{d}\boldsymbol{x}. \tag{6}$$

This regularization term can be rewritten as

$$\int_{\mathcal{D}} Q(\boldsymbol{x}) |\langle \boldsymbol{w}, \boldsymbol{x} \rangle|^2 \mathrm{d}\boldsymbol{x} = \boldsymbol{w}^T H \boldsymbol{w}. \tag{7}$$

where $H(i,j) := \int_{\mathcal{D}} Q(\boldsymbol{x}) x_i x_j \mathrm{d}\boldsymbol{x}$ and $H(i,j)$ denotes element (i,j) of the regularization matrix H. Note that H becomes a positive definite matrix from condition (5).

Combining our regularization approach with the hinge loss function, we propose a classification criterion whereby

$$\text{minimize} \quad \boldsymbol{w}^T H \boldsymbol{w} + c \sum_{n=1}^{N} \xi_n, \tag{8}$$

$$\text{subject to} \quad y_n \left(\langle \boldsymbol{w}, \boldsymbol{x}_n \rangle + b \right) \geq 1 - \xi_n, \tag{9}$$

$$\xi_n \geq 0, \quad n = 1, \ldots, N, \tag{10}$$

where ξ_n are slack variables. The proposed criterion can produce not only various new classifiers, but also a basic SVM by choosing an appropriate weighting function. We demonstrate this in the following sections 3.1 and 3.2. In this

regard, we refer to the proposed classifier as "Weighted Regularization SVM" (WR-SVM).

3.1 Basic Support Vector Machines

In this section, we demonstrate that our regularization criterion produces the basic regularization term (4). In other words, WR-SVM includes basic SVM.

Let us assume that $||\boldsymbol{x}|| = 1$, and $\{x_i, x_j\}(i \neq j)$ are orthogonal. The following assumption holds in the Gaussian kernel model:

$$||\phi(\boldsymbol{x})||^2 = k(\boldsymbol{x}, \boldsymbol{x}) = \exp(-\gamma||\boldsymbol{x} - \boldsymbol{x}||^2) = 1, \tag{11}$$

where $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma||\boldsymbol{x} - \boldsymbol{y}||^2)$ is the Gaussian kernel function. We choose the weighting function to be uniform: $Q_1(\boldsymbol{x}) := S$. Then, the constraint matrix is given by

$$H(i,j) = S\int_{\mathcal{D}} x_i x_j \mathrm{d}\boldsymbol{x} \propto \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}. \tag{12}$$

We can see that the regularization matrix is defined as $H_1 := I_M$, as well as that it is equivalent to (4). Thus, we can regard our regularization method as an extension of the basic regularization term. Also, we can infer that the weighted regularization becomes basic if $Q(\boldsymbol{x})$ is uniform (i.e., no weight).

3.2 Novel Weighted Regularization

Next, we search for an appropriate weighting function. There are two approaches to this, one of which is to make $Q(\boldsymbol{x})$ large in a mixed area of categories. Therefore, we define $Q_2(\boldsymbol{x})$ as a normal distribution:

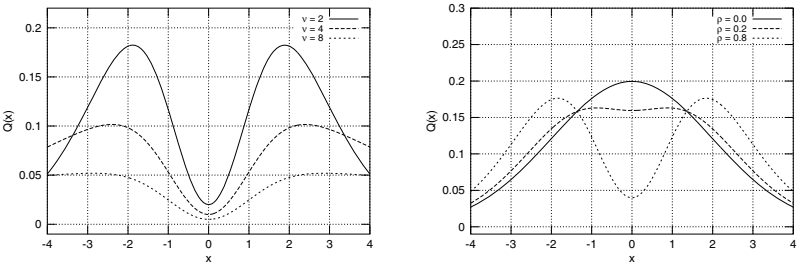
$$Q_2(\boldsymbol{x}) := N(\boldsymbol{x}|\boldsymbol{\mu}, \lambda\Sigma) = \frac{1}{(\sqrt{2\pi})^M\sqrt{|\lambda\Sigma|}} \exp\left(-\frac{1}{2\lambda}(\boldsymbol{x} - \boldsymbol{\mu})\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right), \tag{13}$$

where

$$\boldsymbol{\mu} := \frac{\bar{\boldsymbol{x}}_{+1} + \bar{\boldsymbol{x}}_{-1}}{2}, \quad \Sigma(i,j) := \begin{cases} \frac{1}{N-1}\sum_{n=1}^{N}(\boldsymbol{\mu}(i) - \boldsymbol{x}_n(i))^2 & i = j \\ 0 & i \neq j \end{cases}, \tag{14}$$

and $\bar{\boldsymbol{x}}_{+1}$ and $\bar{\boldsymbol{x}}_{-1}$ denote the mean vectors of labeled patterns $+1$ and -1, respectively. The classifier becomes robust if patterns of different categories are mixed in the central area of the pattern distribution. Furthermore, if we let the parameter λ become sufficiently large, this function becomes similar to a uniform function. Hence, its classifier becomes similar to the basic SVM.

Another approach is to make $Q(\boldsymbol{x})$ small in dense areas and large in sparse areas. Then, this classifier becomes robust for outliers. Thus, we define a weighting function as the difference of two types of normal distribution:



(a) $\rho = 0.9$ is fixed (b) $\nu = 2$ is fixed

Fig. 1. $Q_3(\boldsymbol{x})$: ν and ρ are changed

$$Q_3(\boldsymbol{x}) := \left(1 + \frac{\rho}{\nu^M - 1}\right) N(\boldsymbol{x}|\boldsymbol{\mu}, \nu^2\Sigma) - \frac{\rho}{\nu^M - 1} N(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma), \tag{15}$$

where $0 < \rho < 1$ and $\nu > 1$. If we assume that Σ is a diagonal matrix, then this weighting function always satisfies Eq. (5). Fig 1 depicts examples of such a weighting function. If ν increases, the weighting function becomes smoother and wider. Essentially, ρ should be near 1 (e.g., $\rho = 0.9$), and if ρ is small, the function becomes similar to $Q_2(\boldsymbol{x})$.

The calculation of these regularization matrices includes integration; however, if we use the Gaussian kernel as a basis function, then we can calculate H analytically since $Q(\boldsymbol{x})$ consists of Gauss functions. We present the details of this approach in Section 3.3.

3.3 Analytical Calculation of Regularization Matrices

We define the regularization matrices H_2 and H_3 as

$$H_t(i,j) = \int_{\mathcal{D}} Q_t(\boldsymbol{x}) k(\boldsymbol{x}_i, \boldsymbol{x}) k(\boldsymbol{x}_j, \boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \quad t = 2, 3. \tag{16}$$

Note that it is only necessary to perform integration of the normal distribution and two Gaussian kernel functions analytically. Then, we consider only the following integration:

$$U(i,j) = \int_{\mathcal{D}} N(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma) k(\boldsymbol{x}_i, \boldsymbol{x}) k(\boldsymbol{x}_j, \boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \tag{17}$$

Using the general formula for a Gaussian integral;

$$\int e^{-\frac{1}{2}\boldsymbol{x}^T A \boldsymbol{x} + \boldsymbol{b}\boldsymbol{x}} \mathrm{d}\boldsymbol{x} = \sqrt{\frac{(2\pi)^M}{|A|}} e^{\frac{1}{2}\boldsymbol{b}^T A^{-1} \boldsymbol{b}}, \tag{18}$$

we can calculate Eqs. (17) analytically as follows.

$$U(i,j) = \frac{1}{\sqrt{|4\gamma\Sigma + I_N|}} \exp\left(\frac{1}{2}\boldsymbol{b}_{ij}^T A^{-1}\boldsymbol{b}_{ij} + C_{ij}\right), \tag{19}$$

$$A = 4\gamma I_N + \Sigma^{-1}, \tag{20}$$

$$\boldsymbol{b}_{ij} = 2\gamma(\boldsymbol{x}_i + \boldsymbol{x}_j) + \Sigma^{-1}\boldsymbol{\mu}, \tag{21}$$

$$C_{ij} = -\gamma(||\boldsymbol{x}_i||^2 + ||\boldsymbol{x}_j||^2) - \frac{1}{2}\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu}. \tag{22}$$

H_2 and H_3 can also be calculated in a similar manner.

In practice, the regularization matrix H_t is normalized by $(NH_t)/\mathrm{tr}(H_t)$ so that the adjusting parameter c becomes independence of multiplication factor.

3.4 Novel Classifiers

In this section, we propose novel classifiers by making use of weighted regularization terms.

We assume that the discriminant function is given by

$$D(\boldsymbol{x}|\boldsymbol{\alpha}, b) = \sum_{n=1}^{N} \alpha_n k(\boldsymbol{x}_n, \boldsymbol{x}) + b. \tag{23}$$

Then, the training problem is given by

$$\text{minimize} \quad \frac{1}{2}\boldsymbol{\alpha}^T H_t \boldsymbol{\alpha} + c\sum_{n=1}^{N}\xi_n, \tag{24}$$

$$\text{subject to} \quad y_n\left(\sum_{i=1}^{N}\alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}_n) + b\right) \geq 1 - \xi_n, \tag{25}$$

$$\xi_n \geq 0, \quad n = 1, \ldots, N. \tag{26}$$

We solve this problem by two steps. First, we solve its dual problem:

$$\text{maximize} \quad -\frac{1}{2}\boldsymbol{\beta}^T Y K H_t^{-1} K Y \boldsymbol{\beta} + \sum_{n=1}^{N}\beta_n, \tag{27}$$

$$\text{subject to} \quad 0 \leq \beta_n \leq c, \quad \sum_{n=1}^{N}\beta_n y_n = 0, \quad n = 1, \ldots, N, \tag{28}$$

where $Y := \mathrm{diag}(\boldsymbol{y})$, $\boldsymbol{\beta}$ is a dual parameter vector, and its solution $\hat{\boldsymbol{\beta}}$ can be obtained by quadratic programming [7]. In this regard, a number of quadratic programming solvers have been developed thus far, such as LOQO [1]. Second, the estimated parameters $\hat{\boldsymbol{\alpha}}$ and $\hat{b}$ are given by

$$\hat{\boldsymbol{\alpha}} = H_t^{-1} K Y \hat{\boldsymbol{\beta}}, \qquad \hat{b} = \frac{\mathbf{1}^T\boldsymbol{y} - \boldsymbol{k}^T\hat{\boldsymbol{\alpha}}}{N}. \tag{29}$$

Table 1. UCI Data sets

Name	Training sample	Test samples	Realizations	Dimensions
Banana	400	4900	100	2
B.Cancer	200	77	100	9
Diabetes	468	300	100	8
Flare-Solar	666	400	100	9
German	700	300	100	20
Heart	170	100	100	13
Image	1300	1010	20	18
Ringnorm	400	7000	100	20
Splice	1000	2175	20	60
Thyroid	140	75	100	5
Titanic	150	2051	100	3
Twonorm	400	700	100	20
Waveform	400	4600	100	21

Substituting H_2 or H_3 into Eq.(27), we can construct two novel classifiers. We denote these classifiers as "$\mathrm{SVM_G}$" and "$\mathrm{SVM_D}$", respectively (based on the initials of Gaussian and Difference).

4 Experiments

In this experiment, we used thirteen UCI data sets for binary problems to compare the two novel classifiers $\mathrm{SVM_G}$ and $\mathrm{SVM_D}$ with SVM and L1-norm regularized SVM (L1-SVM).

These data sets are summarized in Table 1, which lists the data set name, the respective numbers of training samples, test samples, realizations, and dimensions.

4.1 Experimental Procedure

Several hyper parameters must be optimized, namely, the kernel parameter γ, the adjusting parameter c, and the weighting parameters λ of $Q_2(\boldsymbol{x})$ and ν of $Q_3(\boldsymbol{x})$, but $\rho = 0.9$ is fixed. These parameters are optimized on the first five realizations of each data set. The best values of each parameter are obtained by using each realization. Finally, the median of the five values is selected. After that, the classifiers are trained and tested for all of the remaining realizations (i.e., 95 or 15 realizations) by using the same parameters.

4.2 Experimental Results

Table 2 contains the results of this experiment. The values in the classifier name column show "average $\pm$ standard deviation" of the error rates for all of the remaining realizations, and the minimum values among all classifiers are marked

Table 2. Experimental results

	λ	SVM_G	L2	L1	ν	SVM_D	L2	L1	SVM	L1-SVM
Banana	1	10.6 ± 0.4	+		8	**10.4 ± 0.4**	+		11.5 ± 0.7	10.5 ± 0.4
B.Cancer	.01	26.3 ± 5.2			4	26.0 ± 4.4			26.0 ± 4.7	**25.4 ± 4.5**
Diabetes	100	24.1 ± 2.0	−	−	8	24.0 ± 2.0	−	−	23.5 ± 1.7	**23.4 ± 1.7**
F.Solar	10	36.7 ± 5.2	−	−	2	**32.4 ± 1.8**		+	**32.4 ± 1.8**	32.9 ± 2.7
German	10	24.0 ± 2.4			16	24.7 ± 2.3			**23.6 ± 2.1**	24.0 ± 2.3
Heart	10	**15.3 ± 3.2**			2	15.6 ± 3.2			16.0 ± 3.3	15.4 ± 3.4
Image	100	4.3 ± 0.9	−		16	4.1 ± 0.8	−	+	**3.0 ± 0.6**	4.8 ± 1.3
Ringnorm	100	**1.5 ± 0.1**	+	+	8	**1.5 ± 0.1**	+	+	1.7 ± 0.1	1.6 ± 0.1
Splice	10	11.3 ± 0.7	−	+	8	11.0 ± 0.5		+	**10.9 ± 0.7**	12.4 ± 0.9
Thyroid	1	7.3 ± 2.9	−	−	2	**4.1 ± 2.0**	+	+	4.8 ± 2.2	5.4 ± 2.4
Titanic	.01	22.4 ± 1.0		+	2	22.5 ± 0.5		+	**22.4 ± 1.0**	23.0 ± 2.1
Twonorm	10	2.4 ± 0.1	+	+	4	**2.3 ± 0.1**	+	+	3.0 ± 0.2	2.7 ± 0.2
Waveform	1	9.7 ± 0.4	+	+	2	**9.6 ± 0.5**	+	+	9.9 ± 0.4	10.1 ± 0.5
Mean %		12.0				**4.2**			6.1	10.7
P-value %		87.2				**71.5**			79.2	87.5

with bold font. The values in the columns for λ and ν show the value selected through model selection for each data set. The signs in columns L2 and L1 show the results of a significance test (t-test with $\alpha = 5\%$) for the differences between SVM_G/SVM_D and SVM/L1-SVM, respectively. "+" indicates that the error obtained with the novel classifier is significantly smaller, while "−" indicates that this error is significantly larger. The penultimate line for "Mean %", is computed by using the average values for all data sets as follows. First, we normalize the error rates by taking

$$\left\{ \frac{(\text{particular value})}{(\text{minimum value})} - 1 \right\} \times 100[\%] \tag{30}$$

for each data set. Next, the "average" values are computed for each classifier. This evaluation method is taken from [8]. The last line shows the average of the p-value between "particular" and "minimum" (i.e., the minimum p-value is 50 %).

SVM_G provides the best results for two data sets. Compared to SVM, SVM_G is significantly better for four data sets and significantly worse for five data sets. Compared to L1-SVM, SVM_G is significantly better for five data sets and significantly worse for three data sets.

Furthermore, SVM_D provides the best results for six data sets. Compared to SVM, SVM_D is significantly better for five data sets and significantly worse for two data sets. Compared to L1-SVM, SVM_D is significantly better for eight data sets and significantly worse for one data sets.

According to the results for both "mean" and "p-value", the SVM_D classifier is the best among the four classifiers considered.

5 Discussion

We showed that the WR-SVM approach includes SVM, and we proposed two novel classifiers ($\mathrm{SVM_G}$ and $\mathrm{SVM_D}$). Furthermore, if the weighting parameters λ and ν are extremely large, then both weighting functions become similar to the uniform distribution. Although both $\mathrm{SVM_G}$ and $\mathrm{SVM_D}$ become similar to the SVM, the regularization matrix H does not become strictly K. Rather,

$$H(i,j) \propto \sqrt{k(\boldsymbol{x}_i, \boldsymbol{x}_j)}. \tag{31}$$

Then, for λ and ν being sufficiently large, neither of the novel classifiers is completely equivalent to SVM. This fact stems from the differences between

$$\int_{\mathcal{D}} Q(\boldsymbol{x}) \left\langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}) \right\rangle^2 \mathrm{d}\boldsymbol{\phi}(\boldsymbol{x}) \text{ and } \int_{\mathcal{D}} Q(\boldsymbol{x}) \left\langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}) \right\rangle^2 \mathrm{d}\boldsymbol{x}. \tag{32}$$

If we switch the weighting functions depending on each data set from among $Q_1(\boldsymbol{x})$, $Q_2(\boldsymbol{x})$ and $Q_3(\boldsymbol{x})$, the classifier will become extremely effective. In fact, $Q_3(\boldsymbol{x})$ coincides with $Q_2(\boldsymbol{x})$ when $\rho = 0$, and since we know that $Q_3(\boldsymbol{x})$ becomes similar to $Q_1(\boldsymbol{x})$ when ν is large, it is possible to choose an appropriate weighting function. However, this increases the number of hyper parameters and makes the model selection problem more difficult.

6 Conclusions and Future Work

In this paper, we proposed both weighted regularization and WR-SVM, and we demonstrated that WR-SVM reduces to the basic SVM upon choosing an appropriate weighting function. This implies that the WR-SVM approach has high general versatility. Furthermore, we proposed two novel classifiers and conducted experiments to compare their performance with existing classifiers. The results demonstrated both the usefulness and the importance of the WR-SVM classifier.

In the future, we plan to improve the performance of the WR-SVM classifier by considering other weighting functions, such as the Gaussian mixture model.

References

1. Benson, H., Vanderbei, R.: Solving problems with semidefinite and related constraints using interior-point methods for nonlinear programming (2002)
2. Bjorck, A.: Numerical methods for least squares problems. Mathematics of Computation (1996)
3. Canu, S., Smola, A.: Kernel methods and the exponential family. Neurocomputing 69, 714–720 (2005)
4. Chen, W.S., Yuen, P., Huang, J., Dai, D.Q.: Kernel machine-based one-parameter regularized fisher discriminant method for face recognition. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 35(4), 659–669 (2005)

5. Huber, P.J.: Robust Statistics. Wiley, New York (1981)
6. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.: Fisher discriminant analysis with kernels. In: Proceedings of the 1999 IEEE Signal Processing Society Workshop Neural Networks for Signal Processing IX, pp. 41–48 (August 1999)
7. Moraru, V.: An algorithm for solving quadratic programming problems. Computer Science Journal of Moldova 5(2), 223–235 (1997)
8. Rätsch, G., Onoda, T., Müller, K.: Soft margins for adaboost. Tech. Rep. NC-TR-1998-021, Royal Holloway College. University of London, UK 42(3), 287–320 (1998)
9. Rennie, J.D.M.: Maximum-margin logistic regression (February 2005), http://people.csail.mit.edu/jrennie/writing
10. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing 14, 199–222 (2004)
11. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society 58(1), 267–288 (1996)
12. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)

Relational Extensions of Learning Vector Quantization

Barbara Hammer, Frank-Michael Schleif, and Xibin Zhu

CITEC center of excellence, Bielefeld University,
33615 Bielefeld, Germany
{bhammer,fschleif,xzhu}@techfak.uni-bielefeld.de

Abstract. Prototype-based models offer an intuitive interface to given data sets by means of an inspection of the model prototypes. Supervised classification can be achieved by popular techniques such as learning vector quantization (LVQ) and extensions derived from cost functions such as generalized LVQ (GLVQ) and robust soft LVQ (RSLVQ). These methods, however, are restricted to Euclidean vectors and they cannot be used if data are characterized by a general dissimilarity matrix. In this approach, we propose relational extensions of GLVQ and RSLVQ which can directly be applied to general possibly non-Euclidean data sets characterized by a symmetric dissimilarity matrix.

Keywords: LVQ, GLVQ, Soft LVQ, Dissimilarity data, Relational data.

1 Introduction

Machine learning techniques have revolutionized the possibility to deal with large electronic data sets by offering powerful tools to automatically learn a regularity underlying the data. However, some of the most powerful machine learning tools which are available today such as the support vector machine act as a black box and their decisions cannot easily be inspected by humans. In contrast, prototype-based methods represent their decisions in terms of typical representatives contained in the input space. Since prototypes can directly be inspected by humans in the same way as data points, an intuitive access to the decision becomes possible: the responsible prototype and its similarity to the given data determine the output.

There exist different possibilities to infer appropriate prototypes from data: Unsupervised learning such as simple k-means, fuzzy-k-means, topographic mapping, neural gas, or the self-organizing map, and statistical counterparts such as the generative topographic mapping infer prototypes based on input data only [1,2,3]. Supervised techniques incorporate class labeling and find decision boundaries which describe priorly known class labels, one of the most popular learning algorithm in this context being learning vector quantization (LVQ) and extensions thereof which are derived from explicit cost functions or statistical models [2,4,5]. Besides different mathematical derivations, these learning algorithms share several fundamental aspects: they represent data in a sparse way

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 481–489, 2011.
© Springer-Verlag Berlin Heidelberg 2011

by means of prototypes, they form decisions based on the similarity of data to prototypes, and training is often very intuitive based on Hebbian principles. In addition, prototype-based models have excellent generalization ability [6,7]. Further, prototypes offer a compact representation of data which can be beneficial for life-long learning, see e.g. the approaches proposed in [8,9,10].

LVQ severely depends on the underlying metric, which is usually chosen as Euclicean metric. Thus, it is unsuitable for complex or heterogeneous data sets where input dimensions have different relevance or a high dimensionality yields to accumulated noise which disrupts the classification. This problem can partially be avoided by appropriate metric learning, see e.g. [7], or by kernel variants, see e.g. [11]. However, if data are inherently non-Euclidean, these techniques cannot be applied. In modern applications, data are often addressed using dedicated non-Euclidean dissimilarities such as dynamic time warping for time series, alignment for symbolic strings, the compression distance to compare sequences based on an information theoretic ground, and similar. These settings do not allow a Euclidean representation of data at all, rather, data are given implicitly in terms of pairwise dissimilarities or relations; we refer to a 'relational data representation' in the following when addressing such settings.

In this contribution, we propose relational extensions of two popular LVQ algorithms derived from cost functions, generalized LVQ (GLVQ) and robust soft LVQ (RSLVQ), respectively [4,5]. This way, these techniques become directly applicable for relational data sets which are characterized in terms of a symmetric dissimilarity matrix only. The key ingredient is taken from recent approaches for relational data processing in the unsupervised domain [12,13]: if prototypes are represented implicitly as linear combinations of data in the so-called pseudo-Euclidean embedding, the relevant distances of data and prototypes can be computed without an explicit reference to a vectorial data representation. This principle holds for every symmetric dissimilarity matrix and thus, allows us to formalize a valid objective of RSLVQ and GLVQ for relational data. Based on this observation, optimization can take place using gradient techniques.

In this contribution, we shortly review LVQ techniques derived from a cost function, and we extend these techniques to relational data. We test the technique on several benchmarks, leading to results comparable to SVM while providing prototype based presentations.

2 Prototype-Based Clustering and Classification

Assume data $\boldsymbol{x}^i \in \mathbb{R}^n, i = 1, \ldots, m$, are given. Prototypes are elements $\boldsymbol{w}^j \in \mathbb{R}^n, j = 1, \ldots, k$, of the same space. They decompose data into receptive fields $R(\boldsymbol{w}^j) = \{\boldsymbol{x}^i \ : \ \forall k \ d(\boldsymbol{x}^i, \boldsymbol{w}^j) \leq d(\boldsymbol{x}^i, \boldsymbol{w}^k)\}$ based on the squared Euclidean distance $d(\boldsymbol{x}^i, \boldsymbol{w}^j) = \|\boldsymbol{x}^i - \boldsymbol{w}^j\|^2$. The goal of prototype-based machine learning techniques is to find prototypes which represent a given data set as accurately as possible.

In supervised settings, data $\boldsymbol{x}^i$ are equipped with class labels $c(\boldsymbol{x}^i) \in \{1, \ldots, L\}$ in a finite set of known classes. Similarly, every prototype is equipped

with a priorly fixed class label $c(\boldsymbol{w}^j)$. A data point is mapped to the class of its closest prototype. The classification error of this mapping is given by the term $\sum_j \sum_{\boldsymbol{x}^i \in R(\boldsymbol{w}^j)} \delta(c(\boldsymbol{x}^i) \neq c(\boldsymbol{w}^j))$ with the delta function δ. This cost function cannot easily be optimized explicitly due to vanishing gradients and discontinuities. Therefore, LVQ relies on a reasonable heuristic by performing Hebbian and unti-Hebbian updates of the prototypes, given a data point [2]. Extensions of LVQ derive similar update rules from explicit cost functions which are related to the classification error, but display better numerical properties such that optimization algorithms can be derived thereof.

Generalized LVQ (GLVQ) has been proposed in the approach [4]. It is derived from a cost function which can be related to the generalization ability of LVQ classifiers [7]. The cost function of GLVQ is given as

$$E_{\mathrm{GLVQ}} = \sum_i \Phi \left(\frac{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) - d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))} \right) \tag{1}$$

where Φ is a differentiable monotonic function such as the hyperbolic tangent, and $\boldsymbol{w}^+(\boldsymbol{x}^i)$ refers to the prototype closest to $\boldsymbol{x}^i$ with the same label as $\boldsymbol{x}^i$, $\boldsymbol{w}^-(\boldsymbol{x}^i)$ refers to the closest prototype with a different label. This way, for every data point, its contribution to the cost function is small if and only if the distance to the closest prototype with a correct label is smaller than the distance to a wrongly labeled prototype, resulting in a correct classification of the point and, at the same time, by optimizing this so-called hypothesis margin of the classifier, aiming at a good generalization ability.

A learning algorithm can be derived thereof by means of a stochastic gradient descent. After a random initialization of prototypes, data $\boldsymbol{x}^i$ are presented in random order. Adaptation of the closest correct and wrong prototype takes place by means of the update rules

$$\Delta \boldsymbol{w}^{\pm}(\boldsymbol{x}^i) \sim \mp \Phi'(\mu(\boldsymbol{x}^i)) \cdot \mu^{\pm}(\boldsymbol{x}^i) \cdot \nabla_{\boldsymbol{w}^{\pm}(\boldsymbol{x}^i)} d(\boldsymbol{x}^i, \boldsymbol{w}^{\pm}(\boldsymbol{x}^i)) \tag{2}$$

where

$$\mu(\boldsymbol{x}^i) = \frac{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) - d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}, \; \mu^{\pm}(\boldsymbol{x}^i) = \frac{2 \cdot d(\boldsymbol{x}^i, \boldsymbol{w}^{\mp}(\boldsymbol{x}^i))}{(d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))^2}. \tag{3}$$

For the squared Euclidean norm, the derivative yields $\nabla_{\boldsymbol{w}^j} d(\boldsymbol{x}^i, \boldsymbol{w}^j) = -2(\boldsymbol{x}^i - \boldsymbol{w}^j)$, leading to Hebbian update rules of the prototypes which take into account the priorly known class information, i.e. they adapt the closest prototypes towards / away from a given data point depending on their labels. GLVQ constitutes one particularly efficient method to adapt the prototypes according to a given labeled data sets.

Robust soft LVQ (RSLVQ) as proposed in [5] constitutes an alternative approach which is based on a statistical model of the data. In the limit of small bandwidth, update rules which are very similar to LVQ result. For non-vanishing bandwidth, soft assignments of data points to prototypes take place. Every prototype induces a probability induced by Gaussians, for example, i.e. $p(\boldsymbol{x}^i|\boldsymbol{w}^j) =$

$K \cdot \exp(-d(\boldsymbol{x}^i, \boldsymbol{w}^j)/2\sigma^2)$ with parameter $\sigma \in \mathbb{R}$ and normalization constant $K = (2\pi\sigma^2)^{-n/2}$. Assuming that every prototype has the same prior, we obtain the overall probability of a data point $p(\boldsymbol{x}^i) = \sum_{\boldsymbol{w}^j} p(\boldsymbol{x}^i|\boldsymbol{w}^j)/k$ and the probability of a point and its corresponding class $p(\boldsymbol{x}^i, c(\boldsymbol{x}^i)) = \sum_{\boldsymbol{w}^j : c(\boldsymbol{w}^j)=c(\boldsymbol{x}^i)} p(\boldsymbol{x}^i|\boldsymbol{w}^j)/k$. The cost function of RSLVQ is given by the quotient

$$E_{\mathrm{RSLVQ}} = \log \prod_i \frac{p(\boldsymbol{x}^i, c(\boldsymbol{x}^i))}{p(\boldsymbol{x}^i)} = \sum_i \log \frac{p(\boldsymbol{x}^i, c(\boldsymbol{x}^i))}{p(\boldsymbol{x}^i)} \tag{4}$$

Considering gradients, we obtain the adaptation rule for every prototype $\boldsymbol{w}^j$ given a training point $\boldsymbol{x}^i$

$$\Delta \boldsymbol{w}^j \sim -\frac{1}{2\sigma^2} \cdot \left(\frac{p(\boldsymbol{x}^i|\boldsymbol{w}^j)}{\sum_{j:c(\boldsymbol{w}^j)=c(\boldsymbol{x}^i)} p(\boldsymbol{x}^i|\boldsymbol{w}^j)} - \frac{p(\boldsymbol{x}^i|\boldsymbol{w}^j)}{\sum_j p(\boldsymbol{x}^i|\boldsymbol{w}^j)} \right) \cdot \nabla_{\boldsymbol{w}^j} d(\boldsymbol{x}^i, \boldsymbol{w}^j) \tag{5}$$

if $c(\boldsymbol{x}^i) = c(\boldsymbol{w}^j)$ and $\Delta \boldsymbol{w}^j \sim \frac{1}{2\sigma^2} \cdot \frac{p(\boldsymbol{x}^i|\boldsymbol{w}^j)}{\sum_j p(\boldsymbol{x}^i|\boldsymbol{w}^j)} \cdot \nabla_{\boldsymbol{w}^j} d(\boldsymbol{x}^i, \boldsymbol{w}^j)$ if $c(\boldsymbol{x}^i) \neq c(\boldsymbol{w}^j)$. Obviously, the scaling factors can be interpreted as soft assignments of the data to corresponding prototypes. The choice of an appropriate parameter σ can critically influence the overall behavior and the quality of the technique, see e.g. [5,14,15] for comparisons of GLVQ and RSLVQ and ways to automatically determine σ based on given data.

3 Dissimilarity Data

In recent years, data are becoming more and more complex in many application domains e.g. due to improved sensor technology or dedicated data formats. To account for this fact, data are often addressed by means of dedicated dissimilarity measures which account for the structural form of the data such as alignment techniques for bioinformatics sequences, dedicated functional norms for mass spectra, the compression distance for texts, etc. Prototype-based techniques such as GLVQ or RSLVQ are restricted to Euclidean vector spaces. Hence their suitability to deal with complex non-Euclidean data sets is highly limited. Prototype-based techniques such as neural gas have recently been extended towards more general data formats [12]. Here we extend GLVQ and RSLVQ to relational variants in a similar way by means of an implicit reference to a pseudo-Euclidean embedding of data.

We assume that data $\boldsymbol{x}^i$ are given as pairwise dissimilarities $d_{ij} = d(\boldsymbol{x}^i, \boldsymbol{x}^j)$. D refers to the corresponding dissimilarity matrix. Note that it is easily possible to transfer similarities to dissimilarities and vice versa, see [13]. We assume symmetry $d_{ij} = d_{ji}$ and we assume $d_{ii} = 0$. However, we do not require that d refers to a Euclidean data space, i.e. D does not need to be embeddable in Euclidean space, nor does it need to fulfill the conditions of a metric.

As argued in [13,12], every such set of data points can be embedded in a so-called pseudo-Euclidean vector space the dimensionality of which is limited by the number of given points. A pseudo-Euclidean vector space is a real-vector

space equipped with the bilinear form $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{p,q} = \boldsymbol{x}^t I_{p,q} \boldsymbol{y}$ where $I_{p,q}$ is a diagonal matrix with p entries 1 and q entries -1. The tuple (p, q) is also referred to as the signature of the space, and the value q determines in how far the standard Euclidean norm has to be corrected by negative eigenvalues to arrive at the given dissimilarity measure. The data set is Euclidean if and only if $q = 0$. For a given matrix D, the corresponding pseudo-Euclidean embedding can be computed by means of an eigenvalue decomposition of the related Gram matrix, which is an $\mathcal{O}(m^3)$ operation. It yields explicit vectors $\boldsymbol{x}^i$ such that $d_{ij} = \langle \boldsymbol{x}^i - \boldsymbol{x}^j, \boldsymbol{x}^i - \boldsymbol{x}^j \rangle_{p,q}$ holds for every pair of data points.

Note that vector operations can be naturally transferred to pseudo-Euclidean space, i.e. we can define prototypes as linear combinations of data in this space. Hence we can perform techniques such as GLVQ explicitly in pseudo-Euclidean space since it relies on vector operations only. One problem of this explicit transfer is given by the computational complexity of the initial embedding, on the one hand, and the fact that out-of-sample extensions to new data points characterized by pairwise dissimilarities are not immediate.

Because of this fact, we are interested in efficient techniques which implicitly refer to such embeddings only. As a side product, such algorithms are invariant to coordinate transforms in pseudo-Euclidean space, rather they depend on the pairwise dissimilarities only instead of the chosen embedding. The key assumption is to restrict prototype positions to linear combination of data points of the form

$$\boldsymbol{w}^j = \sum_i \alpha_{ji} \boldsymbol{x}^i \text{ with } \sum_i \alpha_{ji} = 1 \,. \tag{6}$$

Since prototypes are located at representative points in the data space, it is a reasonable assumption to restrict prototypes to the affine subspace spanned by the given data points. In this case, dissimilarities can be computed implicitly by means of the formula

$$d(\boldsymbol{x}^i, \boldsymbol{w}^j) = [D \cdot \alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D \alpha_j \tag{7}$$

where $\alpha_j = (\alpha_{j1}, \ldots, \alpha_{jn})$ refers to the vector of coefficients describing the prototype $\boldsymbol{w}^j$ implicitly, as shown in [12].

This observation constitutes the key to transfer GLVQ and RSLVQ to relational data without an explicit embedding in pseudo-Euclidean space. Prototype $\boldsymbol{w}^j$ is represented implicitly by means of the coefficient vectors α_j. Then, we can use the equivalent characterization of distances in the GLVQ and RSVLQ cost function leading to the costs of relational GLVQ (RGLVQ) and relational RSLVG (RSLVQ), respectively:

$$E_{\text{RGLVQ}} = \sum_i \Phi \left(\frac{[D\alpha^+]_i - \frac{1}{2} \cdot (\alpha^+)^t D \alpha^+ - [D\alpha^-]_i + \frac{1}{2} \cdot (\alpha^-)^t D \alpha^-}{[D\alpha^+]_i - \frac{1}{2} \cdot (\alpha^+)^t D \alpha^+ + [D\alpha^-]_i - \frac{1}{2} \cdot (\alpha^-)^t D \alpha^-} \right), \tag{8}$$

where as before the closest correct and wrong prototype are referred to, corresponding to the coefficients α^+ and α^-, respectively. A stochastic gradient

descent leads to adaptation rules for the coefficients α^+ and α^- in relational GLVQ: component k of these vectors is adapted as

$$\Delta\alpha_k^{\pm} \sim \mp \Phi'(\mu(\boldsymbol{x}^i)) \cdot \mu^{\pm}(\boldsymbol{x}^i) \cdot \frac{\partial \left([D\alpha^{\pm}]_i - \frac{1}{2} \cdot (\alpha^{\pm})^t D\alpha^{\pm}\right)}{\partial \alpha_k^{\pm}} \tag{9}$$

where $\mu(\boldsymbol{x}^i)$, $\mu^+(\boldsymbol{x}^i)$, and $\mu^-(\boldsymbol{x}^i)$ are as above. The partial derivative yields

$$\frac{\partial \left([D\alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D\alpha_j\right)}{\partial \alpha_{jk}} = d_{ik} - \sum_l d_{lk}\alpha_{jl} \tag{10}$$

Similarly,

$$E_{\text{RRSLVQ}} = \sum_i \log \frac{\sum_{\alpha_j : c(\alpha_j)=c(\boldsymbol{x}^i)} p(\boldsymbol{x}^i|\alpha_j)/k}{\sum_{\alpha_j} p(\boldsymbol{x}^i|\alpha_j)/k} \tag{11}$$

where $p(\boldsymbol{x}^i|\alpha_j) = K \cdot \exp\left(-\left([D\alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D\alpha_j\right)/2\sigma^2\right)$. A stochastic gradient descent leads to the adaptation rule

$$\Delta\alpha_{jk} \sim -\frac{1}{2\sigma^2} \cdot \left(\frac{p(\boldsymbol{x}^i|\alpha_j)}{\sum_{j:c(\alpha_j)=c(\boldsymbol{x}^i)} p(\boldsymbol{x}^i|\alpha_j)} - \frac{p(\boldsymbol{x}^i|\alpha_j)}{\sum_j p(\boldsymbol{x}^i|\alpha_j)} \right) \cdot \frac{\partial \left([D\alpha_j]_i - \frac{1}{2}\alpha_j^t D\alpha_j\right)}{\partial \alpha_{jk}} \tag{12}$$

if $c(\boldsymbol{x}^i) = c(\alpha_j)$ and $\Delta\alpha_{jk} \sim \frac{1}{2\sigma^2} \cdot \frac{p(\boldsymbol{x}^i|\alpha_j)}{\sum_j p(\boldsymbol{x}^i|\alpha_j)} \cdot \frac{\partial\left([D\alpha_j]_i - \frac{1}{2}\alpha_j^t D\alpha_j\right)}{\partial \alpha_{jk}}$ if $c(\boldsymbol{x}^i) \neq c(\alpha_j)$. After every adaptation step, normalization takes place to guarantee $\sum_i \alpha_{ji} = 1$.

The prototypes are initialized as random vectors, i.e we initialize α_{ij} with small random values such that the sum is one. It is possible to take class information into account by setting all α_{ij} to zero which do not correspond to the class of the prototype. The prototype labels can then be determined based on their receptive fields before adapting the initial decision boundaries by means of supervised learning vector quantization.

An extension of the classification to new data is immediate based on an observation made in [12]: given a novel data point $\boldsymbol{x}$ characterized by its pairwise dissimilarities $D(\boldsymbol{x})$ to the data used for training, the dissimilarity of $\boldsymbol{x}$ to a prototype represented by α_j is $d(\boldsymbol{x}, \boldsymbol{w}^j) = D(\boldsymbol{x})^t \cdot \alpha_j - \frac{1}{2} \cdot \alpha_j^t D\alpha_j$.

Note that, for GLVQ, a kernelized version has been proposed in [11]. However, this refers to a kernel matrix only, i.e. it requires Euclidean similarities instead of general symmetric dissimilarities. In particular, it must be possible to embed data in a possibly high dimensional Euclidean feature space. Here we extended GLVQ and RSLVQ to relational data characterized by a general symmetric dissimilarities which might be induced by strictly non-Euclidean data.

4 Experiments

We evaluate the algorithms for several benchmark data sets where data are characterized by pairwise dissimilarities. On the one hand, we consider six data

Table 1. Results of prototype based classification in comparison to SVM for diverse dissimilarity data sets. The classification accuracy obtained in a repeated cross-validation is reported, the standard deviation is given in parenthesis. SVM results marked with * are taken from [16]. For Cat Cortex, Vibrio, Chromosome, the respective best SVM result is reported by using different preprocessing mechanisms clip, flip, shift, and similarities as features with linear and Gaussian kernel.

	#Data Points	#Labels	**RGLVQ**	**RRSLVQ**	**best SVM**	**#Proto.**
Amazon47	204	47	0.81(0.01)	**0.83**(0.02)	0.82*	94
Aural Sonar	100	2	**0.88**(0.02)	0.85(0.02)	0.87*	10
Face Rec.	945	139	**0.96**(0.00)	**0.96**(0.00)	**0.96***	139
Patrol	241	8	0.84(0.01)	0.85(0.01)	**0.88***	24
Protein	213	4	0.92(0.02)	0.53(0.01)	**0.97***	20
Voting	435	2	**0.95**(0.01)	0.62(0.01)	**0.95***	20
Cat Cortex	65	5	0.93(0.01)	0.94(0.01)	**0.95**	12
Vibrio	4200	22	**1.00**(0.00)	0.94(0.08)	**1.00**	49
Chromosome	1100	49	0.93(0.00)	0.80(0.01)	**0.95**	63

sets used also in [16]: Amazon47, Aural-Sonar, Face Recognition, Patrol, Protein and Voting. In additional we consider the Cat Cortex from [18], the Copenhagen Chromosomes data [17] and one own data set, the Vibrio data, which consists of 1,100 samples of vibrio bacteria populations characterized by mass spectra. The spectra contain approx. 42,000 mass positions. The full data set consists of 49 classes of vibrio-sub-species. The preprocessing of the Vibrio data is described in [20] and the underlying similarity measures in [21,20]. The article [16] investigates the possibility to deal with similarity/dissimilarity data which is non-Euclidean with the SVM. Since the corresponding Gram matrix is not positive semidefinite, according preprocessing steps have to be done which make the SVM well defined. These steps can change the spectrum of the Gram matrix or they can treat the dissimilarity values as feature vectors which can be processed by means of a standard kernel.

Since some of these matrices correspond to similarities rather than dissimilarities, we use standard preprocessing as presented in [13]. For every data set, a number of prototypes which mirrors the number of classes was used, representing every class by only few prototypes relating to the choices as taken in [12], see Tab. 1. The evaluation of the results is done by means of the classification accuracy as evaluated on the test set in a ten fold repeated cross-validation (nine tenths of date set for training, one tenth for testing) with ten repeats. The results are reported in Tab. 1. In addition, we report the best results obtained by SVM after diverse preprocessing techniques [16].

Interestingly, in most cases, results which are comparable to the best SVM as reported in [16] can be found, whereby making preprocessing as done in [16] superfluous. Further, unlike for SVM which is based on support vectors in the data set, solutions are represented as typical prototypes.

5 Conclusions

We have presented an extension of prototype-based techniques to general possibly non-Euclidean data sets by means of an implicit embedding in pseudo-Euclidean data space and a corresponding extension of the cost function of GLVQ and RSLVQ to this setting. As a result, a very powerful learning algorithm can be derived which, in most cases, achieves results which are comparable to SVM but without the necessity of according preprocessing since relational LVQ can directly deal with possibly non-Euclidean data whereas SVM requires a positive semidefinite Gram matrix. Similar to SVM, relational LVQ has quadratic complexity due to its dependency on the full dissimilarity matrix. A speed-up to linear techniques e.g. by means of the Nyström approximation for dissimilarity data similar to [22] is the subject of ongoing research.

Acknowledgement. Financial support from the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative and from the "German Science Foundation (DFG)" under grant number HA-2719/4-1 is gratefully acknowledged.

References

1. Martinetz, T.M., Berkovich, S.G., Schulten, K.J.: 'Neural-gas' Network for Vector Quantization and Its Application to Time-series Prediction. IEEE Trans. on Neural Networks 4(4), 558–569 (1993)
2. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer-Verlag New York, Inc. (2001)
3. Bishop, C., Svensen, M., Williams, C.: The Generative Topographic Mapping. Neural Computation 10(1), 215–234 (1998)
4. Sato, A., Yamada, K.: Generalized Learning Vector Quantization. In: Proceedings of the 1995 Conference Advances in Neural Information Processing Systems, vol. 8, pp. 423–429. MIT Press, Cambridge (1996)
5. Seo, S., Obermayer, K.: Soft Learning Vector Quantization. Neural Computation 15(7), 1589–1604 (2003)
6. Hammer, B., Villmann, T.: Generalized Relevance Learning Vector Quantization. Neural Networks 15(8-9), 1059–1068 (2002)
7. Schneider, P., Biehl, M., Hammer, B.: Adaptive Relevance Matrices in Learning Vector Quantization. Neural Computation 21(12), 3532–3561 (2009)
8. Denecke, A., Wersing, H., Steil, J.J., Koerner, E.: Online Figure-Ground Segmentation with Adaptive Metrics in Generalized LVQ. Neurocomputing 72(7-9), 1470–1482 (2009)
9. Kietzmann, T., Lange, S., Riedmiller, M.: Incremental GRLVQ: Learning Relevant Features for 3D Object Recognition. Neurocomputing 71(13-15), 2868–2879 (2008)
10. Alex, N., Hasenfuss, A., Hammer, B.: Patch Clustering for Massive Data Sets. Neurocomputing 72(7-9), 1455–1469 (2009)
11. Qin, A.K., Suganthan, P.N.: A Novel Kernel Prototype-based Learning Algorithm. In: Proc. of ICPR 2004, pp. 621–624 (2004)
12. Hammer, B., Hasenfuss, A.: Topographic Mapping of Large Dissimilarity Data Sets. Neural Computation 22(9), 2229–2284 (2010)

13. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore (2005)
14. Schneider, P., Biehl, M., Hammer, B.: Hyperparameter Learning in Probabilistic Prototype-based Models. Neurocomputing 73(7-9), 1117–1124 (2010)
15. Seo, S., Obermayer, K.: Dynamic Hyperparameter Scaling Method for LVQ Algorithms. In: IJCNN, pp. 3196–3203 (2006)
16. Chen, Y., Eric, K.G., Maya, R.G., Ali, R.L.C.: Similarity-based Classification: Concepts and Algorithms. Journal of Machine Learning Research 10, 747–776 (2009)
17. Neuhaus, M., Bunke, H.: Edit Distance Based Kernel functions for Structural Pattern Classification. Pattern Recognition 39(10), 1852–1863 (2006)
18. Haasdonk, B., Bahlmann, C.: Learning with Distance Substitution Kernels. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 220–227. Springer, Heidelberg (2004)
19. Lundsteen, C., Phillip, J., Granum, E.: Quantitative Analysis of 6985 Digitized Trypsin g-banded Human Metaphase Chromosomes. Clinical Genetics 18, 355–370 (1980)
20. Maier, T., Klebel, S., Renner, U., Kostrzewa, M.: Fast and Reliable maldi-tof ms–based Microorganism Identification. Nature Methods 3 (2006)
21. Barbuddhe, S.B., Maier, T., Schwarz, G., Kostrzewa, M., Hof, H., Domann, E., Chakraborty, T., Hain, T.: Rapid Identification and Typing of Listeria Species by Matrix-assisted Laser Desorption Ionization-time of Flight Mass Spectrometry. Applied and Environmental Microbiology 74(17), 5402–5407 (2008)
22. Gisbrecht, A., Hammer, B., Schleif, F.-M., Zhu, X.: Accelerating Dissimilarity Clustering for Biomedical Data Analysis. In: Proceedings of SSCI (2011)

On Low-Rank Regularized Least Squares for Scalable Nonlinear Classification

Zhouyu Fu, Guojun Lu, Kai-Ming Ting, and Dengsheng Zhang

Gippsland School of IT, Monash University, Churchill, VIC 3842, Australia
{zhouyu.fu,guojun.lu,kaiming.ting,dengsheng.zhang}@infotech.monash.edu.au

Abstract. In this paper, we revisited the classical technique of Regularized Least Squares (RLS) for the classification of large-scale nonlinear data. Specifically, we focus on a low-rank formulation of RLS and show that it has linear time complexity in the data size only and does not rely on the number of labels and features for problems with moderate feature dimension. This makes low-rank RLS particularly suitable for classification with large data sets. Moreover, we have proposed a general theorem for the closed-form solutions to the Leave-One-Out Cross Validation (LOOCV) estimation problem in empirical risk minimization which encompasses all types of RLS classifiers as special cases. This eliminates the reliance on cross validation, a computationally expensive process for parameter selection, and greatly accelerate the training process of RLS classifiers. Experimental results on real and synthetic large-scale benchmark data sets have shown that low-rank RLS achieves comparable classification performance while being much more efficient than standard kernel SVM for nonlinear classification. The improvement in efficiency is more evident for data sets with higher dimensions.

Keywords: Classification, Regularized Least Squares, Low-Rank Approximation.

1 Introduction

Classification is a fundamental problem in data mining. It involves learning a function that separates data points from different classes. The support vector machine (SVM) classifier, which aims at recovering a maximal margin separating hyperplane in the feature space, is a powerful tool for classification and has demonstrated state-of-the-art performance in many problems [1]. SVM can operate directly in the input space by finding linear decision boundaries. Despite its simplicity, linear SVM is quite restricted in discriminative power and can not handle linearly inseparable data. This limits its applicability to nonlinear problems arising in real-world applications. We can also learn a SVM in the feature space via the kernel trick which leads to nonlinear decision boundaries. The kernel SVM has better classification performance than linear SVM, but its scalability is an issue for large-scale nonlinear classification. Despite the existence of faster SVM solvers like LibSVM [2], training of kernel SVM is still time

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 490–499, 2011.
© Springer-Verlag Berlin Heidelberg 2011

consuming for moderately large data sets. Linear SVM training, however, can be made very fast [3,4] due to its different problem structure. It would be much desirable to have a classification tool that achieves the best of the two worlds with the performance of nonlinear SVM while scaling well to larger data sets.

In this paper, we examine Regularized Least Squares (RLS) as an alternative to SVM in the setting of large-scale nonlinear classification. To this end, we focus on a low-rank formulation of RLS initially proposed in [5]. The paper makes the following contributions to low-rank RLS. Firstly, we have empirically investigated the performance of low-rank RLS for large-scale nonlinear classification with real data sets. It can be observed from the empirical results that low-rank RLS achieves comparable performance to nonlinear SVM while being much more efficient. Secondly, as suggested by our computational analysis and evidenced by experimental results, low-rank RLS has linear time complexity in the data size only and independent of the feature dimension and number of class labels. This property makes low-rank RLS particularly suited to multi-class problems with many class labels and moderate feature dimensions. Thirdly, we also propose a theorem on the closed-form estimation for Leave-One-Out-Cross-Validation (LOOCV) under mild conditions. This includes RLS as special cases and provides the LOOCV estimation for low-rank RLS. Consequently, we can avoid the time consuming step for choosing classifier parameters using k-fold cross validation, which involves classifier training and testing on different data partitions k times for each parameter setting. With the proposed theorem, we can obtain exact prediction results of LOOCV by training the classifier with the specified parameters only once. This greatly reduces the time spent on the selection of classifier parameters.

2 Classification with Regularized Least Squares Classifier

In this section, we present the RLS classifier. We focus on binary classification, since multiclass problems can be converted to binary ones using decomposition schemes [6]. In a binary classification problem, a i.i.d. training sample $\{\mathbf{x}_i, y_i | i = 1, \ldots, N\}$ of size N is randomly drawn from some unknown but fixed distribution $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$, where $\mathcal{X} \subset \mathbb{R}^d$ is the feature space with dimension d and $\mathcal{Y} = \{-1, 1\}$ specify the labels. The purpose is to design a classifier function $f : \mathcal{X} \to \mathcal{Y}$ that can best predict the labels of novel test data drawn from the same distribution. This can usually be achieved by solving the following Empirical Risk Minimization (ERM) problem [1]

$$\min_f L(f) = \lambda\Omega(f) + \sum_i \ell(y_i, f(\mathbf{x}_i)) \tag{1}$$

where the first term on the right-hand-side is the regularization term for the classifier function $f(.)$, and the second term is the empirical risk over the training instances. $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}^+$ is the loss function correlated with classification error.

The ERM problem in Equation 1 specifies a general framework for classifier learning. Depending on different forms of the loss function ℓ, different types of

classifiers can be derived based on the above formulation. Two widely used loss functions, namely the hinge loss for SVM and the squared loss for RLS, are listed below.

$$\ell(y_i, f_i) = \max(0, 1 - y_i f_i) \qquad \text{Hinge Loss (SVM)} \tag{2}$$

$$\ell(y_i, f_i) = (y_i - f_i)^2 = (1 - y_i f_i)^2 \qquad \text{Square Loss (RLS)} \tag{3}$$

where $f_i \triangleq f(\mathbf{x}_i)$ denotes the decision value for $\mathbf{x}_i$. The minor difference in the loss functions of SVM and RLS lead to very different routines for optimization. Closed-form solutions can be obtained for RLS, whereas the optimization problem for SVM is much harder and remains an active research topic in machine learning [3,4].

Consider the linear RLS classifier with linear decision function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. The general ERM problem defined in 1 reduces to

$$\min_{\mathbf{w}} \lambda\|\mathbf{w}\|^2 + \sum_i (\mathbf{w}^T\mathbf{x}_i - y_i)^2 \tag{4}$$

The weight vector $\mathbf{w}$ can be obtained in closed-form by

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{5}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$ is the data matrix formed by input features in rows, $\mathbf{y} = [y_1, \ldots, y_N]^T$ the column vector of binary label variables, and $\mathbf{I}$ is an identify matrix.

The ERM formulation can also be used to solve nonlinear classification problems. In the nonlinear case, the classifier function $f(.)$ is defined over the domain of Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$. An RKHS $\mathcal{H}$ is a Hilbert space associated with a kernel function $\kappa : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$. The kernel explicitly defines the inner product between two vectors in RKHS, i.e. $\kappa(\mathbf{x}_i, \mathbf{x}) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle$ with $\phi(.) \in \mathcal{H}$. We can think of $\phi(\mathbf{x}_i)$ as a mapping of the input feature vector $\mathbf{x}$ in the RKHS. In the linear case, $\phi(\mathbf{x}) = \mathbf{x}$. In the nonlinear case, the explicit form of the mapping ϕ is unknown but the inner product is well defined by the kernel κ. Let $\Omega(f) = \|f\|^2_{\mathcal{H}}$ be the regularization term for f in RKHS, according to the representer theorem [1], the solution of Equation 1 takes the following solution

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) \tag{6}$$

Let $\alpha = [\alpha_1, \ldots, \alpha_N]$ be a vector of coefficients, and $\mathbf{K} \in \mathbb{R}^{N \times N}$ be the Gram matrix whose (i, j)th entry stores the kernel evaluation for input examples $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e. $\mathbf{K}_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. The regularization term becomes

$$\Omega(f) = \|f\|^2_{\mathcal{H}} = \alpha^T\mathbf{K}\alpha$$

The optimization problem for RLS can then be formulated by

$$\min_{\alpha} \lambda\alpha^T\mathbf{K}\alpha + \sum_i \|\mathbf{K}\alpha - \mathbf{y}\|^2 \tag{7}$$

The solution of α is

$$\alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y} \tag{8}$$

The classifier function is in the form of Equation 6 with above α.

3 Low-Rank Regularized Least Squares

3.1 Low-Rank Approximation for RLS

It can be seen from Equation 8 that the main computation of RLS is the inversion of the $N \times N$ kernel matrix $\mathbf{K}$, which depends on the size of the data set. For large-scale data sets with many training examples, it is infeasible to solve the above equation directly.

A low-rank formulation of RLS first proposed in [5] can be derived to tackle the larger data sets. The idea is quite straightforward. Instead of taking a full expansion of kernel function values over all training instances in Equation 6, we can take a subset of them leading to a reduced representation for the classifier function $f(.)$

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}_i, \mathbf{x}) \tag{9}$$

Without loss of generality, we assume that the first m instances are selected to form the above expansion with $m \ll N$. The RLS problem arising from the above representation of classifier function $f(.)$ is given by

$$\min_{\alpha} L(\alpha) = \lambda \alpha^T \mathbf{K}_{\mathcal{S},\mathcal{S}} \alpha + \|\mathbf{K}_{\mathcal{X},\mathcal{S}} \alpha - \mathbf{y}\|^2 \tag{10}$$

where $\alpha = [\alpha_1, \ldots, \alpha_m]^T$ is a vector of m coefficients for the selected prototypes, and is much smaller than the full N-dimensional coefficient vector in standard kernel RLS. $\mathbf{K}_{\mathcal{S},\mathcal{S}}$ is the $m \times m$ submatrix at the top-left corner of the big matrix $\mathbf{K}$, and $\mathbf{K}_{\mathcal{X},\mathcal{S}}$ is a $N \times m$ matrix by taking the first m columns from matrix $\mathbf{K}$. The above-defined low-rank RLS problem has the following closed-form solution

$$\alpha = (\mathbf{K}_{\mathcal{X},\mathcal{S}}^T \mathbf{K}_{\mathcal{X},\mathcal{S}} + \lambda \mathbf{K}_{\mathcal{S},\mathcal{S}})^{-1} \mathbf{K}_{\mathcal{X},\mathcal{S}}^T \mathbf{y} \tag{11}$$

This only involves the inversion of a $m \times m$ matrix and is much more efficient than inverting a $N \times N$ matrix. The classifier function $f(.)$ for low-rank RLS has the simple form below

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) \tag{12}$$

3.2 Time Complexity Analysis

The three most time-consuming operations for solving Equation 11 are the evaluation of the reduced kernel matrix $\mathbf{K}_{\mathcal{X},\mathcal{S}}$, the matrix product $\mathbf{K}_{\mathcal{X},\mathcal{S}}^T \mathbf{K}_{\mathcal{X},\mathcal{S}}$, and the

inverse of $\mathbf{K}_{\mathcal{X},\mathcal{S}}^T\mathbf{K}_{\mathcal{X},\mathcal{S}} + \lambda\mathbf{K}_{\mathcal{S},\mathcal{S}}$. The complexity of kernel evaluation is $O(Nmd)$, which depends on the data size N, the subset size m and feature dimension d. The matrix product takes $O(Nm^2)$ time to compute, and the inverse has a complexity of $O(m^3)$ for $m \times m$ square matrix. Since $m \ll N$, the complexity of the inverse is dominated by that of matrix product. Besides, normally we have the relation $d < m$ for classification problems with moderate dimensions[1]. Thus, the computation of Equation 11 is largely determined by the the calculation of matrix product $\mathbf{K}_{\mathcal{X},\mathcal{S}}^T\mathbf{K}_{\mathcal{X},\mathcal{S}}$ with complexity of $O(Nm^2)$, which scales linearly with the size of the training data set given fixed m and does not depend on the dimension of the data. Besides, low-rank RLS also scales well to increasing number of labels. Each additional label just increase the complexity by $O(Nm)$, which is trivial compared to the expensive operations described above.

3.3 Closed-Form LOOCV Estimation

Another important problem is in the selection of the regularization parameter λ in RLS (Equations 5, 8 and 11). The standard way to do so is Cross Validation (CV) by splitting the training data sets into k folds and repeating training and testing k times. Each time using one fold data as the validation set and the remaining data for training. The performance is evaluated on each validation set for each CV round and candidate parameter value of λ. This could be quite time consuming for larger k values and a large search range for the parameter. In this subsection, we introduce a theorem for obtaining closed-form estimation for LOOCV under mild conditions, i.e. the case for $k = N$ where each training instance is used once as the singleton validation set. The theorem provides a way to estimate LOOCV solution for low-rank in closed form by learning just a single classifier on the whole training data set without retraining. It also includes standard RLS classifiers as special cases.

Let $\mathbf{Z}^{\sim j}$ denote the jth Leave-One-Out (LOO) sample by removing the jth instance $\mathbf{z}_j = \{\mathbf{x}_j, y_j\}$ from the full data set $\mathbf{Z}$. Let $f(.) = \arg\min_f L(f|\mathbf{Z}, \ell)$ and $f^{\sim j}(.) = \arg\min_f L(f|\mathbf{Z}^{\sim j}, \ell)$ be the minimizers of the RLS problems for $\mathbf{Z}$ and $\mathbf{Z}^{\sim j}$ respectively. The LOOCV estimation on the training data is obtained by $f^{\sim j}(\mathbf{x}_j)$ for each j. The purpose here is to find a solution to $f^{\sim j}(\mathbf{x}_j)$ directly from f without retraining the classifier for each LOO sample $\mathbf{Z}^{\sim j}$. This is not possible for arbitrary loss functions ℓ and general forms of function f. However, if ℓ and f satisfy certain conditions, it is possible to obtain a closed-form solution to LOOCV estimation.

We now show the main theorem for LOOCV estimation in the following

Theorem 1. *Let f be the solution to the ERM problem in Equation 7 for a random sample $\mathbf{Z} = \{\mathbf{X}, \mathbf{y}\}$. If the prediction vector $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]$ can be expressed in the form $\mathbf{f} = \mathbf{H}\mathbf{y}$, and the loss function $\ell(f(\mathbf{x}), y) = 0$ whenever*

[1] We have fixed $m = 1000$ for all our experiments in this paper. With $m = 1000$, we expect a feature dimension in the order of 100 or smaller would not much contribute to the time complexity compared to the calculation of $\mathbf{K}_{\mathcal{X},\mathcal{S}}^T\mathbf{K}_{\mathcal{X},\mathcal{S}}$.

$f(\mathbf{x}) = y$, then the LOOCV estimate for the jth data point $\mathbf{x}_j$ in the training set is given by

$$f^{\sim j}(\mathbf{x}_j) = \frac{f(\mathbf{x}_j) - H_{j,j} y_j}{1 - H_{j,j}} \tag{13}$$

Proof.

$$\begin{aligned} L(f^{\sim j} | \mathbf{Z}^{\sim j}, \ell) &= \sum_{i \neq j} \ell(y_i, f^{\sim j}(\mathbf{x}_i)) + \lambda \Omega(f) \\ &= \sum_{i} \ell(y_i^j, f^{\sim j}(\mathbf{x}_i)) + \lambda \Omega(f) \end{aligned} \tag{14}$$

where $\mathbf{y}^j = [y_1^j, \ldots, y_N^j]$ with $y_i^j = y_i$ for $i \neq j$ and $y_j^j = f^{\sim j}(\mathbf{x}_j)$. The second equality is true due to that $\ell(y_j^j, f^{\sim j}(\mathbf{x}_j)) = 0$. Hence $f^{\sim j}$ is also the solution to the ERM problem on training data $\mathbf{X}$ with modified label vector $\mathbf{y}^j$. Let $\mathbf{f}^{\sim j}$ be the solution vector for $f^{\sim j}(.)$. By the linearity assumption, we have $\mathbf{f}^{\sim j} = \mathbf{H}\mathbf{y}^j$ and $\mathbf{f} = \mathbf{H}\mathbf{y}$. The LOOCV estimate for the jth instance $f^{\sim j}(\mathbf{x}_j)$ is given by the jth component of the solution vector $\mathbf{f}^{\sim j}$, i.e. $f^{\sim j}(\mathbf{x}_j) \triangleq f_j^{\sim j}$. The following relation holds for $f_j^{\sim j}$

$$\begin{aligned} f_j^{\sim j} &= \sum_i H_{j,i} y_i^{\sim j} = \sum_{i \neq j} H_{j,i} y_i^{\sim j} + H_{j,j} y_j^{\sim j} \\ &= \sum_{i \neq j} H_{j,i} y_i + H_{j,j} f_j^{\sim j} = f_j - H_{j,j} y_j + H_{j,j} f_j^{\sim j} \end{aligned} \tag{15}$$

where $f_j \triangleq f(\mathbf{x}_j)$ is the decision value for $\mathbf{x}_j$ returned by $f(.)$. This leads to

$$f_j^{\sim j} = \frac{f_j - H_{j,j} y_j}{1 - H_{j,j}}$$

The loss function for RLS satisfies the identity relation $\ell(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2 = 0$ whenever $f(\mathbf{x}) = y$. The solution of RLS can also be expressed by the linear form over the label vector. Different variations of RLS can take slightly different forms of $\mathbf{H}$ in Equation 13, which is listed in Table 1. The closed-form LOOCV estimations for linear RLS and kernel RLS discussed in [5] are special cases of the theorem. Besides, the theorem also provides the closed-form solution to LOOCV for the low-rank RLS, which has not yet been discovered.

Table 1. Summary of different RLS solutions and $\mathbf{H}$ matrices

RLS Type	Weight Vector $\mathbf{w}$	Prediction	$\mathbf{H}$
Linear	$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$	$\mathbf{X}\mathbf{w}$	$\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T$
Kernel	$(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}$	$\mathbf{K}\mathbf{w}$	$\mathbf{K}(\mathbf{K} + \lambda\mathbf{I})^{-1}$
Low Rank	$(\mathbf{K}_{\mathcal{X},\mathcal{S}}^T\mathbf{K}_{\mathcal{X},\mathcal{S}} + \lambda\mathbf{K}_{\mathcal{S},\mathcal{S}})^{-1}\mathbf{K}_{\mathcal{S},\mathcal{S}}\mathbf{y}$	$\mathbf{K}_{\mathcal{X},\mathcal{S}}\mathbf{w}$	$\mathbf{K}_{\mathcal{X},\mathcal{S}}(\mathbf{K}_{\mathcal{X},\mathcal{S}}^T\mathbf{K}_{\mathcal{X},\mathcal{S}} + \lambda\mathbf{K}_{\mathcal{S},\mathcal{S}})^{-1}\mathbf{K}_{\mathcal{S},\mathcal{S}}$

4 Experimental Results

In this section, we describe the experiments performed to demonstrate the performance of the RLS classifier for the classification of large-scale nonlinear data sets and experimentally validate the claims established for RLS in the previous section about its linear-time complexity and closed form LOOCV estimation.

The experiments were conducted on 8 large data sets chosen from the UCI machine learning repository [7], and 2 multi-label classification data sets (tmc2007 and mediamill) chosen from the MULAN repository [8]. Table 2 gives a brief summary of the data sets used, such as number of labels, feature dimension, the sizes of training and testing sets for each data set. Due to the large sizes

Table 2. Summary of data sets used for experiments

	satimage	usps	letter	tmc2007	mediamill	connect-4	shuttle	ijcnn1	mnist	SensIT
Labels	6	10	26	22	50	3	7	2	10	3
Dimension	36	256	16	500	120	126	9	22	778	100
Training Size	4435	7291	15000	21519	30993	33780	43500	49990	60000	78823
Testing Size	2000	2007	5000	7077	12914	33777	14500	91701	10000	19705

of these data sets, standard RLS is infeasible here and the low-rank RLS has been used instead throughout our experiments. We simply refer the low-rank RLS as RLS hereafter. The subset of prototypes $\mathcal{S}$ were randomly chosen from the training instances and used to compute matrices $\mathbf{K}_{\mathcal{X},\mathcal{S}}$ and $\mathbf{K}_{\mathcal{S},\mathcal{S}}$ in Equation 11. We have found that random selection of prototypes has performed well empirically. For each data set, we also applied standard kernel and linear SVM classifiers to compare their performances to RLS. The LibSVM package [2] was used to train kernel SVMs, which implements the SMO algorithm [9] for fast SVM training. For both kernel SVM and RLS, we have used the Gaussian kernel function $\kappa(\mathbf{x}, \mathbf{z}) = \exp(-g\|\mathbf{x} - \mathbf{z}\|)$ where g is empirically set to the inverse of feature dimension. The feature values are standardized to have zero mean and unit norm for each dimension before kernel computation and classifier training are applied. The LibLinear package [4] was used to train linear SVMs in the primal formulation. We have adopted the one-vs-all framework to tackle both multi-class and multi-label data by training a binary classifier for each class label to distinguish from other labels. The training and testing of each classification algorithm was repeated 10 times for each data set. The Areas Under ROC Curve (AUC) value was used as the performance measure for classification for two reasons. Firstly, AUC is a metric commonly used for both standard and multi-label classification problems. More importantly, AUC is an aggregate measure for classification performance which takes into consideration the full range of the ROC curve. In contrast, alternative measures like the error rate simply counts the number of misclassified examples corresponding to a single point on the ROC curve. This may lead to an over-estimation of classification performance for imbalanced problems, where classification error is largely determined by the performance on the dominant class. For multiclass problems, the

average AUC value over all class labels was used for performance comparison. The means and standard deviations of AUC values over different testing rounds achieved by RLS, linear (LSVM) and kernel SVMs are reported in Table 3. The average CPU time spent on a single training round for each method and data set is also included in the same table.

Table 3. Performance comparison of RLS with kernel and linear SVMs in terms of accuracy in prediction and efficiency in training

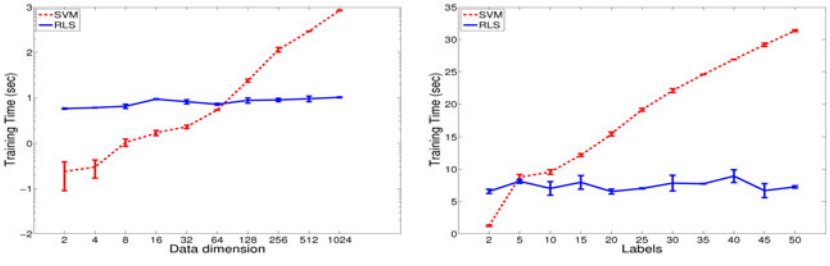
Dataset	AUC			Time (sec)		
	RLS	SVM	LSVM	RLS	SVM	LSVM
satimage	0.985 ± 0.002	0.986 ± 0.001	0.925 ± 0.002	6.0 ± 0.4	2.3 ± 0.1	0.2 ± 0.0
usps	0.997 ± 0.002	0.998 ± 0.002	0.987 ± 0.004	7.9 ± 0.5	26.8 ± 0.5	8.3 ± 0.3
letter	0.998 ± 0.000	0.999 ± 0.000	0.944 ± 0.001	10.2 ± 0.8	24.8 ± 0.4	1.2 ± 0.0
tmc2007	0.929 ± 0.003	0.927 ± 0.005	0.925 ± 0.014	20.0 ± 1.4	289.6 ± 48.8	128.7 ± 15.0
mediamill	0.839 ± 0.012	0.807 ± 0.020	0.827 ± 0.008	23.8 ± 1.8	3293.7 ± 370.6	181.2 ± 6.1
connect-4	0.861 ± 0.002	0.895 ± 0.001	0.813 ± 0.001	22.1 ± 1.1	1783.1 ± 145.1	1.9 ± 0.1
shuttle	0.999 ± 0.002	0.979 ± 0.029	0.943 ± 0.009	26.0 ± 1.0	7.1 ± 0.2	0.6 ± 0.0
ijcnn1	0.994 ± 0.000	0.997 ± 0.000	0.926 ± 0.005	29.1 ± 0.7	38.1 ± 2.7	0.3 ± 0.0
mnist	0.994 ± 0.000	0.999 ± 0.000	0.985 ± 0.000	53.9 ± 7.5	16256.1 ± 400.5	211.5 ± 3.7
SensIT	0.934 ± 0.001	0.939 ± 0.001	0.918 ± 0.001	48.6 ± 9.8	9588.4 ± 471.8	5.5 ± 0.2

From Table 3, we can see that RLS is highly competitive with SVM in classification performance while being more efficient. This is especially true for large data sets with higher dimensions and/or multiple labels. For most data sets, the performance gap between the two methods is small. On the other hand, linear SVM, although being very efficient, does not achieve satisfactory performances and is outperformed by both RLS and SVM by a large gap on most data sets. The comparison results presented here clearly shows the potential of RLS for the classification of large-scale nonlinear data.

Another interesting observation we can make from Table 3 is the linear-time complexity of RLS with respect to the size of training set only. The rows in the table are actually arranged in increasing order by the size of the training set, which is monotonically related to the training time of RLS displayed in the 5th column of the same table. The training time of RLS is not much influenced by the number of labels as well as the feature dimension of the classification problem. This is apparently not the case for SVM and LSVM, which spend more time on problems with more labels and larger number of features, like mnist.

To better see the point that RLS has superior scalability as compared to SVM for higher dimensional data and multiple labels, we have further performed two experiments on synthetic data sets. In the first experiment, we simulate a binary classification setting by randomly generating data points from two separate Gaussian distributions in d dimensional Euclidean space. By varying the value d from 2 to 1024 incremented by the power of 2, we trained SVM and RLS classifiers for 10 random samples of size 10000 for each d and recorded the training times in seconds. The training times are plotted in log scale against the

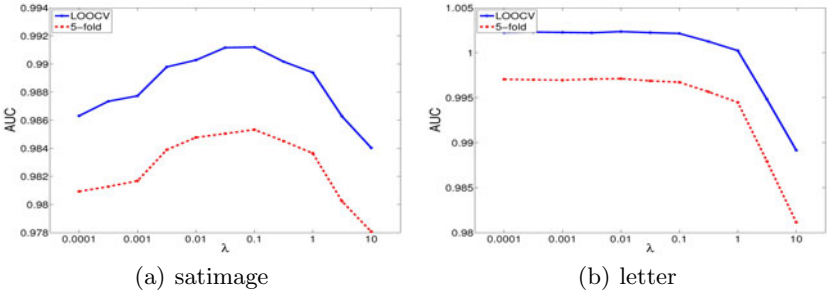
d values in Figure 1(a). From the figure, we can see that SVM is much faster than RLS initially for smaller values of d, but the training time increases dramatically with growing dimensions. RLS, on the other hand, scales surprisingly well to higher data dimensions, which have little effect on the training speed of RLS as can be seen from the figure. In Figure 1(b), we show the training times against increasing number of labels by fixing $d = 8$, where data points were generated from a separate Gaussian model for each label. Not surprisingly, we can see that increasing number of labels has little effect on training speed for RLS.



(a) (b)

Fig. 1. Comparison of training speed for SVM and RLS with (a) growing data dimensions; (b) increasing number of classes. Solid line shows the training time in seconds for RLS, and broken line shows the time for SVM.

In our final experiment, we validate the proposed closed-form LOOCV estimation for RLS. To this end, we have compared the AUC value calculated from LOOCV estimation with that obtained from a separate 5-fold cross validation process for each candidate parameter value of λ. Figure 2 shows the plots of AUC values returned by the two different processes against varying λ values. As can be seen from the plots, the curves returned by closed-form LOOCV estimations (in solid lines) are quite consistent with those returned by the empirical CV processes (in broken lines). Similar trends can be observed from the two curves in most subfigures. However, it involves classifier training only once for LOOCV



(a) satimage (b) letter

Fig. 2. Comparison of cross validation performance for closed-form LOOCV and 5-fold CV. LOOCV curves are offset by 0.005 in the vertical direction for clarity.

by using the closed-form estimation, whereas classifier training and testing need be repeated k times for the empirical k-fold cross validation. In the worst case, this can be about k times as expensive as the analytic solution.

5 Conclusions

We examined low-rank RLS classifier in the setting of large-scale nonlinear classification, which achieves comparable performance with kernel SVM but scales much better to larger data sizes, higher feature dimensions and increasing number of labels. Low-rank RLS has much potential for different classification applications. One possibility is to apply it to multi-label classification by combining it with various label transformation methods proposed for multi-label learning which is likely to produce many subproblems with the same data and different labels [8].

Acknowledgments. This work was supported by the Australian Research Council under the Discovery Project (DP0986052) entitled "Automatic music feature extraction, classification and annotation".

References

1. Scholkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2002)
2. Fan, R.E., Chen, P.H., Lin, C.J.: Working set selection using the second order information for training SVM. Journal of Machine Learning Research 6, 1889–1918 (2005)
3. Joachims, T.: Training linear SVMs in linear time. In: SIGKDD (2006)
4. Hsieh, C.-J., Chang, K.W., Lin, C.J., Keerthi, S., Sundararajan, S.: A dual coordinate descent method for large-scale linear SVM. In: Intl. Conf. on Machine Learning (2008)
5. Rifkin, R.: Everything Old Is New Again: A Fresh Look at Historical Approaches. PhD thesis, Mass. Inst. of Tech (2002)
6. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. Journal of Machine Learning Research 5, 101–141 (2004)
7. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
8. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data Data Mining and Knowledge Discovery Handbook, pp. 667–685 (2010)
9. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods - Support Vector Learning. MIT Press (1998)

Multitask Learning Using Regularized Multiple Kernel Learning

Mehmet Gönen[1], Melih Kandemir[1], and Samuel Kaski[1,2]

[1] Aalto University School of Science
Department of Information and Computer Science
Helsinki Institute for Information Technology HIIT
[2] University of Helsinki
Department of Computer Science
Helsinki Institute for Information Technology HIIT

Abstract. Empirical success of kernel-based learning algorithms is very much dependent on the kernel function used. Instead of using a single fixed kernel function, *multiple kernel learning* (MKL) algorithms learn a combination of different kernel functions in order to obtain a similarity measure that better matches the underlying problem. We study *multitask learning* (MTL) problems and formulate a novel MTL algorithm that trains coupled but nonidentical MKL models across the tasks. The proposed algorithm is especially useful for tasks that have different input and/or output space characteristics and is computationally very efficient. Empirical results on three data sets validate the generalization performance and the efficiency of our approach.

Keywords: kernel machines, multilabel learning, multiple kernel learning, multitask learning, support vector machines.

1 Introduction

Given a sample of N independent and identically distributed training instances $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where $\boldsymbol{x}_i$ is a D-dimensional input vector and y_i is its target output, kernel-based learners find a decision function in order to predict the target output of an unseen test instance $\boldsymbol{x}$ [10,11]. For example, the decision function for binary classification problems (i.e., $y_i \in \{-1, +1\}$) can be written as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

where the kernel function ($k\colon \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$) calculates a similarity metric between data instances. Selecting the kernel function is the most important issue in the training phase; it is generally handled by choosing the best-performing kernel function among a set of kernel functions on a separate validation set.

In recent years, *multiple kernel learning* (MKL) methods have been proposed [4], for learning a combination k_η of multiple kernels instead of selecting one:

$$k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\eta}) = f_\eta(\{k_m(\boldsymbol{x}_i, \boldsymbol{x}_j)_{m=1}^P\}; \boldsymbol{\eta})$$

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 500–509, 2011.
© Springer-Verlag Berlin Heidelberg 2011

where the combination function ($f_\eta : \mathbb{R}^P \rightarrow \mathbb{R}$) forms a single kernel from P base kernels using the parameters $\boldsymbol{\eta}$. Different kernels correspond to different notions of similarity and instead of searching which works best, the MKL method does the picking for us, or may use a combination of kernels. MKL also allows us to combine different representations possibly from different sources or modalities.

When there are multiple related machine learning problems, tasks or data sets, it is reasonable to assume that also the models are related and to learn them jointly. This is referred to as *multitask learning* (MTL). If the input and output domains of the tasks are the same (e.g., when modeling different users of the same system as the tasks), we can train a single learner for all the tasks together. If the input and/or output domains of the tasks are different (e.g., in multilabel classification where each task is defined as predicting one of the labels), we can share the model parameters between the tasks while training.

In this paper, we formulate a novel algorithm for *multitask multiple kernel learning* (MTMKL) that enables us to train a single learner for each task, benefiting from the generalization performance of the overall system. We learn similar kernel functions for all of the tasks using separate but regularized MKL parameters, which corresponds to using a similar distance metric for each task. We show that such coupled training of MKL models across the tasks is better than training MKL models separately on each task, referred to as *single-task multiple kernel learning* (STMKL).

In Section 2, we give an overview of the related work. Section 3 explains the key properties of the proposed algorithm. We then demonstrate the performance of our MTMKL method on three data sets in Section 4. We conclude by a summary of the general aspects of our contribution in Section 5.

We use the following notation throughout the rest of this paper. We use boldface lowercase letters to denote vectors and boldface uppercase letters to denote matrices. The i and j are used as indices for the training instances, r and s for the tasks, and m for the kernels. The T and P are the numbers of the tasks and the kernels to be combined, respectively. The number of training instances in task r is denoted by N^r.

2 Related Work

[2] introduces the idea of multitask learning, in the sense of learning related tasks together by sharing some aspects of the task-specific models between all the tasks. The ultimate target is to improve the performance of each individual task by exploiting the partially related data points of other tasks.

The most frequently used strategy for extending discriminative models to multitask learning is by following the hierarchical Bayes intuition of ensuring similarity in parameters across the tasks by binding the parameters of separate tasks [1]. Parameter binding typically involves a coefficient to tune the similarity between the parameters of different tasks. This idea is introduced to kernel-based algorithms by [3]. In essence, they achieve parameter similarity by decomposing

the hyperplane parameters into shared and task-specific components. The model reduces to a single-kernel learner with the following kernel function:

$$\widehat{k}(\boldsymbol{x}_i^r, \boldsymbol{x}_j^s) = (1/\nu + \delta_r^s) k(\boldsymbol{x}_i^r, \boldsymbol{x}_j^s)$$

where ν determines the similarity between the parameters of different tasks and δ_r^s is 1 if $r = s$ and 0 otherwise. The same model can be extended to MKL using a combined kernel function:

$$\widehat{k_\eta}(\boldsymbol{x}_i^r, \boldsymbol{x}_j^s; \boldsymbol{\eta}) = (1/\nu + \delta_r^s) k_\eta(\boldsymbol{x}_i^r, \boldsymbol{x}_j^s; \boldsymbol{\eta}) \tag{1}$$

where we can learn the combination parameters $\boldsymbol{\eta}$ using standard MKL algorithms. This task-dependent kernel approach has three disadvantages: (a) It requires all tasks to be in a common input space to be able to calculate the kernel function between the instances of different tasks. (b) It requires all tasks to have similar target outputs to be able to capture them in a single learner. (c) It requires more time than training separate but small learners for each task.

There are some recent attempts to integrate MTL and MKL in multilabel settings. [5] uses multiple hypergraph kernels with shared parameters across the tasks to learn multiple labels of a given data set together. Learning the large set of kernel parameters in this special case of the multilabel setup requires a computationally intensive learning procedure. In a similar study, [12] suggests decomposing the kernel weights into shared and label-specific components. They develop a computationally feasible, but still intensive, algorithm for this model. In a multitask setting, [9] proposes to use the same kernel weights for each task.

[6] proposes a feature selection method that uses separate hyperplane parameters for the tasks and joins them by regularizing the weights of each feature over the tasks. This method enforces the tasks to use each feature either in all tasks or in none. [7] uses the parameter sharing idea to extend the large margin nearest neighbor classifier to multitask learning by decomposing the covariance matrix of the Mahalanobis metric into task-specific and task-independent parts. They report that using different but similar distance metrics for the tasks increases generalization performance.

Instead of binding different tasks using a common learner as in [3], we propose a general and computationally efficient MTMKL framework that binds the different tasks to each other through the MKL parameters, which is discussed under multilabel learning setup by [12]. They report that using different kernel weights for each label does not help and suggest to use a common set of weights for all labels. We allow the tasks to have their own learners in order to capture the task-specific properties and to use similar kernel functions (i.e., separate but regularized MKL parameters), which corresponds to using similar distance metrics as in [7], in order to capture the task-independent properties.

3 Multitask Learning Using Multiple Kernel Learning

There are two possible approaches to integrate MTL and MKL under a general and computationally efficient framework: (a) using common MKL parameters

for each task, and (b) using separate MKL parameters but regularizing them in order to have similar kernel functions for each task. The first approach is also discussed in [9] and we use this approach as a baseline comparison algorithm.

Sharing exactly the same set of kernel combination parameters might be too restrictive for weakly correlated tasks. Instead of using the same kernel function, we can learn different kernel combination parameters for each task and regularize them to obtain similar kernels. Model parameters can be learned jointly by solving the following min-max optimization problem:

$$\underset{\{\boldsymbol{\eta}^r \in \mathcal{E}\}_{r=1}^T}{\text{mininimize}} \; \mathcal{O}_\eta = \left\{ \underset{\{\boldsymbol{\alpha}^r \in \mathcal{A}^r\}_{r=1}^T}{\text{maximize}} \; \Omega(\{\boldsymbol{\eta}^r\}_{r=1}^T) + \sum_{r=1}^T J^r(\boldsymbol{\alpha}^r, \boldsymbol{\eta}^r) \right\} \tag{2}$$

where $\Omega(\cdot)$ is the regularization term calculated on the kernel combination parameters, the $\mathcal{E}$ denotes the domain of the kernel combination parameters, $J^r(\cdot,\cdot)$ is the objective function of the kernel-based learner of task r, which is generally composed of a regularization term and an error term, and the $\mathcal{A}^r$ is the domain of the parameters of the kernel-based learner of task r.

If the tasks are binary classification problems (i.e., $y_i^r \in \{-1, +1\}$) and the squared error loss is used implying least squares support vector machines, the objective function and the domain of the model parameters of task r become

$$J^r(\boldsymbol{\alpha}^r, \boldsymbol{\eta}^r) = \sum_{i=1}^{N^r} \alpha_i^r - \frac{1}{2} \sum_{i=1}^{N^r} \sum_{j=1}^{N^r} \alpha_i^r \alpha_j^r y_i^r y_j^r \left(k_\eta^r(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r; \boldsymbol{\eta}^r) + \frac{\delta_i^j}{2C} \right)$$

$$\mathcal{A}^r = \left\{ \boldsymbol{\alpha}^r \colon \sum_{i=1}^{N^r} \alpha_i^r y_i^r = 0, \; \alpha_i^r \in \mathbb{R} \;\; \forall i \right\}$$

where C is the regularization parameter. If the tasks are regression problems (i.e., $y_i^r \in \mathbb{R}$) and the squared error loss is used implying kernel ridge regression, the objective function and the domain of the model parameters of task r are

$$J^r(\boldsymbol{\alpha}^r, \boldsymbol{\eta}^r) = \sum_{i=1}^{N^r} \alpha_i^r y_i^r - \frac{1}{2} \sum_{i=1}^{N^r} \sum_{j=1}^{N^r} \alpha_i^r \alpha_j^r \left(k_\eta^r(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r; \boldsymbol{\eta}^r) + \frac{\delta_i^j}{2C} \right)$$

$$\mathcal{A}^r = \left\{ \boldsymbol{\alpha}^r \colon \sum_{i=1}^{N^r} \alpha_i^r = 0, \; \alpha_i^r \in \mathbb{R} \;\; \forall i \right\}.$$

If we use a convex combination of kernels, the domain of the kernel combination parameters becomes

$$\mathcal{E} = \left\{ \boldsymbol{\eta} \colon \sum_{m=1}^{P} \eta_m = 1, \; \eta_m \geq 0 \;\; \forall m \right\}$$

and the combined kernel function of task r with the convex combination rule is

$$k_\eta^r(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r; \boldsymbol{\eta}^r) = \sum_{m=1}^{P} \eta_m^r k_m^r(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r).$$

Similarity between the combined kernels is enforced by adding an explicit regularization term to the objective function. We propose the sum of the dot products between kernel combination parameters as the regularization term:

$$\Omega(\{\boldsymbol{\eta}^r\}_{r=1}^T) = -\nu \sum_{r=1}^{T} \sum_{s=1}^{T} \langle \boldsymbol{\eta}^r, \boldsymbol{\eta}^s \rangle. \tag{3}$$

Using a very small ν value corresponds to treating the tasks as unrelated, whereas a very large value enforces the model to use similar kernel combination parameters across the tasks. The regularization function can also be interpreted as the negative of the total correlation between the kernel weights of the tasks and we want to minimize the negative of the total correlation if the tasks are related. Note that the regularization function is concave but efficient optimization is possible thanks to the bounded feasible sets of the kernel weights.

The min-max optimization problem in (2) can be solved using an alternating optimization procedure analogous to many MKL algorithms in the literature [8,13,14]. Algorithm 1 summarizes the training procedure. First, we initialize the kernel combination parameters $\{\boldsymbol{\eta}^r\}_{r=1}^T$ uniformly. Given $\{\boldsymbol{\eta}^r\}_{r=1}^T$, the problem reduces to training T single-task single-kernel learners. After training these learners, we can update $\{\boldsymbol{\eta}^r\}_{r=1}^T$ by performing a projected gradient-descent steps to order to satisfy two constraints on the kernel weights: (a) being positive and (b) summing up to one. For faster convergence, this update procedure can be interleaved with a line search method (e.g., Armijo's rule) to pick the step sizes at each iteration. These two steps are repeated until convergence, which can be checked by monitoring the successive objective function values.

Algorithm 1. Multitask Multiple Kernel Learning with Separate Parameters

1: Initialize $\boldsymbol{\eta}^r$ as $\left(1/P \ \ldots \ 1/P\right)^\top \quad \forall r$
2: **repeat**
3: Calculate $\mathbf{K}_\eta^r = \left\{k_\eta^r(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r; \boldsymbol{\eta}^r)\right\}_{i,j=1}^{N^r} \quad \forall r$
4: Solve a single-kernel machine using $\mathbf{K}_\eta^r \quad \forall r$
5: Update $\boldsymbol{\eta}^r$ in the opposite direction of $\partial \mathcal{O}_\eta / \partial \boldsymbol{\eta}^r \quad \forall r$
6: **until** convergence

If the kernel combination parameters are regularized with the function (3), in the binary classification case, the gradients with respect to $\boldsymbol{\eta}^r$ are

$$\frac{\partial \mathcal{O}_\eta}{\partial \eta_m^r} = -2\nu \sum_{s=1}^{T} \eta_m^s - \frac{1}{2} \sum_{i=1}^{N^r} \sum_{j=1}^{N^r} \alpha_i^r \alpha_j^r y_i^r y_j^r k_m^r(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)$$

and, in the regression case,

$$\frac{\partial \mathcal{O}_\eta}{\partial \eta_m^r} = -2\nu \sum_{s=1}^{T} \eta_m^s - \frac{1}{2} \sum_{i=1}^{N^r} \sum_{j=1}^{N^r} \alpha_i^r \alpha_j^r k_m^r(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r).$$

4 Experiments

We test the proposed MTMKL algorithm on three data sets. We implement the algorithm and baseline methods, altogether one STMKL and three MTMKL algorithms, in MATLAB[1]. `STMKL` learns separate STMKL models for each task. `MTMKL(R)` is the MKL variant of regularized MTL model of [3], outlined in (11). `MTMKL(C)` is the MTMKL model that has common kernel combination parameters across the tasks, outlined in [9]. `MTMKL(S)` is the new MTMKL model that has separate but regularized kernel combination parameters across the tasks, outlined in Algorithm 1.

We use the squared error loss for both classification and regression problems. The regularization parameters C and ν are selected using cross-validation from $\{0.01, 0.1, 1, 10, 100\}$ and $\{0.0001, 0.01, 1, 100, 10000\}$, respectively. For each data set, we use the same cross-validation setting (i.e., the percentage of data used in training and the number of folds used for splitting the training data) reported in the previous studies to have directly comparable results.

4.1 Cross-Platform siRNA Efficacy Data Set

The cross-platform small interfering RNA (siRNA) efficacy data set[2] contains 653 siRNAs targeted on 52 genes from 14 cross-platform experiments with corresponding 19 features. We combine 19 linear kernels calculated on each feature separately. Each experiment is treated as a separate task and we use ten random splits where 80 per cent of the data is used for training. We apply two-fold cross-validation on the training data to choose regularization parameters.

Table 1. Root mean squared errors on the cross-platform siRNA data set

Method	RMSE
`STMKL`	23.89 ± 0.97
`MTMKL(R)`	37.66 ± 2.38
`MTMKL(C)`	23.53 ± 1.05
`MTMKL(S)`	23.45 ± 1.05

Table 1 gives the root mean squared error for each algorithm. `MTMKL(R)` is outperformed by all other algorithms because the target output spaces of the experiments are very different. Hence, training a separate learner for each cross-platform experiment is more reasonable. `MTMKL(C)` and `MTMKL(S)` are both better than `STMKL` in terms of the average performance, and `MTMKL(S)` is statistically significantly better (the paired t-test with the confidence level $\alpha = 0.05$).

[1] Implementations are available at `http://users.ics.tkk.fi/gonen/mtmkl`
[2] Available at `http://lifecenter.sgst.cn/RNAi`

4.2 MIT Letter Data Set

The MIT letter data set[3] contains 8×16 binary images of handwritten letters from over 180 different writers. A multitask learning problem, which has eight binary classification problems as its tasks, is constructed from the following pairs of letters and the number of data instances for each task is given in parentheses: {a,g} (6506), {a,o} (7931), {c,e} (7069), {f,t} (3057), {g,y} (3693), {h,n} (5886), {i,j} (5102), and {m,n} (6626). We combine five different kernels on binary feature vectors: the linear kernel and the polynomial kernel with degrees 2, 3, 4, and 5. We use ten random splits where 50 per cent of the data of each task is used for training. We apply three-fold cross-validation on the training data to choose regularization parameters. Note that MTMKL(R) cannot be trained for this problem because the output domains of the tasks are different.

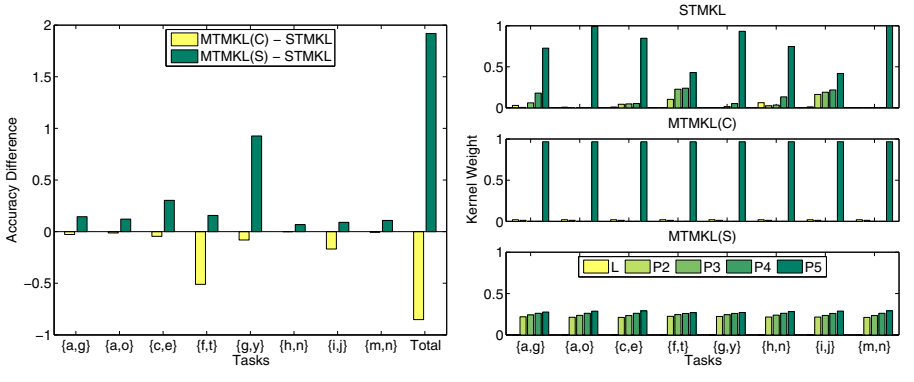


Fig. 1. Comparison of the three algorithms on the MIT letter data set. **Left:** Average accuracy differences. **Right:** Average kernel weights.

Figure 1 shows the average accuracy differences of MTMKL(C) and MTMKL(S) over STMKL. We see that MTMKL(S) consistently improves classification accuracy compared to STMKL and the improvement is statistically significant on six out of eights tasks (the paired t-test with the confidence level $\alpha = 0.05$), whereas MTMKL(C) could not improve classification accuracy on any of the tasks and it is statistically significantly worse on two tasks. Figure 1 also gives the average kernel weights of STMKL, MTMKL(C), and MTMKL(S). We see that STMKL and MTMKL(C) use the fifth degree polynomial kernel with very high weights, whereas MTMKL(S) uses all four polynomial kernels with nearly equal weights.

4.3 Cognitive State Inference Data Set

Finally, we evaluate the algorithms on a multilabel setting where each label is regarded as a task. The learning problem is to infer latent affective and cognitive states of a computer user based on physiological measurements. In the

[3] Available at http://www.cis.upenn.edu/~taskar/ocr

experiments, we measure six male users with four sensors (an accelerometer, a single-line EEG, an eye tracker, and a heart-rate sensor) while they are shown 35 web pages that include a personal survey, several preference questions, logic puzzles, feedback to their answers, and some instructions, one for each page. After the experiment, they are asked to annotate their cognitive state over three numerical Likert scales (valence, arousal, and cognitive load). Our features consist of summary measures of the sensor signals extracted from each page. Hence, our data set consisted of $6 \times 35 = 210$ data points and three output labels for each. We combine four Gaussian kernels on feature vectors of each sensor separately. We use ten random splits where 75 per cent of the data of each task is used for training. We apply three-fold cross-validation on the training data to choose regularization parameters. Note that `MTMKL(R)` cannot be applied to multilabel classification.

Learning inference models of this kind, which predict the cognitive and emotional state of the user, has a central role in cognitive user interface design. In such setups, a major challenge is that the training labels are inaccurate and scarce because collecting them is laborious to the users.

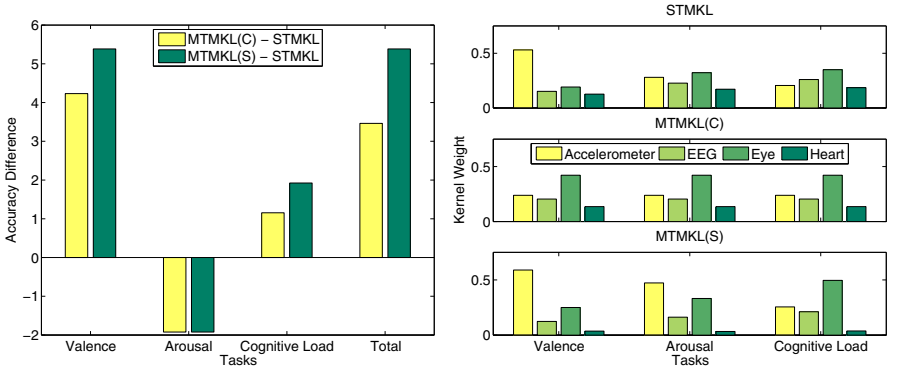


Fig. 2. Comparison of the three algorithms on the cognitive state inference data set. **Left:** Average accuracy differences. **Right:** Average kernel weights.

Figure 2 shows the accuracy differences of `MTMKL(C)` and `MTMKL(S)` over `STMKL` and reveals that learning and predicting the labels jointly helps to eliminate the noise present in the labels. Two of the three output labels (valence and cognitive load) are predicted more accurately in a multitask setup, with a positive change in the total accuracy. Note that `MTMKL(S)` is better than `MTMKL(C)` at predicting these two labels, and they perform equally well for the remaining one (arousal). Figure 2 also gives the kernel weights of `STMKL`, `MTMKL(C)`, and `MTMKL(S)`. We see that `STMKL` assigns very different weights to sensors for each label, whereas `MTMKL(C)` obtains better classification performance using the same weights across labels. `MTMKL(S)` assigns kernel weights between these two extremes and further increases the classification performance. We also see that the features extracted

from the accelerometer are more informative than the other features for predicting valence; likewise, eye tracker is more informative for predicting cognitive load.

4.4 Computational Complexity

Table 2 summarizes the average running times of the algorithms on the data sets used. Note that `MTMKL(R)` and `MTMKL(S)` need to choose two parameters, C and ν, whereas `STMKL` and `MTMKL(C)` choose only C in the cross-validation phase. `MTMKL(R)` uses the training instances of all tasks in a single learner and always requires significantly more time than the other algorithms. We also see that `STMKL` and `MTMKL(C)` take comparable times and `MTMKL(S)` takes more time than these two because of the longer cross-validation phase.

Table 2. Running times of the algorithms in seconds

Data Set	`STMKL`	`MTMKL(R)`	`MTMKL(C)`	`MTMKL(S)`
Cross-Platform siRNA Efficacy	7.14	114.88	4.78	16.17
MIT Letter	9211.60	NA	8847.14	18241.32
Cognitive State Inference	5.23	NA	3.32	20.53

5 Conclusions

In this paper, we introduce a novel multiple kernel learning algorithm for multitask learning. The proposed algorithm uses separate kernel weights for each task, regularized to be similar. We show that training using a projected gradient-descent method is efficient. Defining the interaction between tasks to be over kernel weights instead of over other model parameters allows learning multitask models even when the input and/or output characteristics of the tasks are different. Empirical results on several data sets show that the proposed method provides high generalization performance with reasonable computational cost.

Acknowledgments. The authors belong to the Adaptive Informatics Research Centre (AIRC), a Center of Excellence of the Academy of Finland. This work was supported by the Nokia Research Center (NRC) and in part by the Pattern Analysis, Statistical Modeling and Computational Learning (PASCAL2), a Network of Excellence of the European Union.

References

1. Baxter, J.: A Bayesian/information theoretic model of learning to learn via multiple task sampling. Machine Learning 28(1), 7–39 (1997)
2. Caruana, R.: Multitask learning. Machine Learning 28(1), 41–75 (1997)

3. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: Kim, W., Kohavi, R., Gehrke, J., DuMouchel, W. (eds.) Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 109–117. ACM (2004)
4. Gönen, M., Alpaydın, E.: Multiple kernel learning algorithms. Journal of Machine Learning Research 12, 2211–2268 (2011)
5. Ji, S., Sun, L., Jin, R., Ye, J.: Multi-label multiple kernel learning. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems, vol. 21, pp. 777–784. MIT Press (2009)
6. Obozinski, G., Taskar, B., Jordan, M.I.: Joint covariate selection and joint subspace selection for multiple classification problems. Statistics and Computing 20(2), 231–252 (2009)
7. Parameswaran, S., Weinberger, K.Q.: Large margin multi-task metric learning. In: Lafferty, J., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A. (eds.) Advances in Neural Information Processing Systems, vol. 23, pp. 1867–1875. MIT (2010)
8. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: SimpleMKL. Journal of Machine Learning Research 9, 2491–2521 (2008)
9. Rakotomamonjy, A., Flamary, R., Gasso, G., Canu, S.: $\ell_p - \ell_q$ penalty for sparse linear and sparse multiple kernel multi-task learning. IEEE Transactions on Neural Networks 22(8), 1307–1320 (2011)
10. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2002)
11. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York (2004)
12. Tang, L., Chen, J., Ye, J.: On multiple kernel learning with multiple labels. In: Boutilier, C. (ed.) Proceedings of the 21st International Joint Conference on Artifical Intelligence, pp. 1255–1260 (2009)
13. Varma, M., Babu, B.R.: More generality in efficient multiple kernel learning. In: Danyluk, A.P., Bottou, L., Littman, M.L. (eds.) Proceedings of the 26th International Conference on Machine Learning, p. 134. ACM (2009)
14. Xu, Z., Jin, R., Yang, H., King, I., Lyu, M.R.: Simple and efficient multiple kernel learning by group Lasso. In: Fürnkranz, J., Joachims, T. (eds.) Proceedings of the 27th International Conference on Machine Learning, pp. 1175–1182. Omnipress (2010)

Solving Support Vector Machines beyond Dual Programming

Xun Liang

School of Information, Renmin University of China, Beijing 100872, China
xliang@ruc.edu.cn

Abstract. Support vector machines (SV machines, SVMs) are solved conventionally by converting the convex primal problem into a dual problem with the aid of a Lagrangian function, during whose process the non-negative Lagrangian multipliers are mandatory. Consequently, in the typical C-SVMs, the optimal solutions are given by stationary saddle points. Nonetheless, there may still exist solutions beyond the stationary saddle points. This paper explores these new points violating Karush-Kuhn-Tucker (KKT) condition.

Keywords: Support vector machines, Generalized Lagrangian function, Commonwealth SVMs, Commonwealth points, Stationary saddle points, Singular points, KKT condition.

1 Introduction

Support vector machines (SV machines, SVMs) training involves a convex optimization problem, and SVMs' solutions are solved at stationary points. However, affiliated SVM architectures could still possibly have negative or out-of-upper-bound configurations, sometimes found at non-stationary points. However, for optimal solutions at non-stationary points and/or outside the first quadrant or beyond the upper bound, most literature neither provided any justification nor furnished techniques to approach to optimal and equally applicable solutions for SVMs. For a purpose of safer applications, the geometrical structure of optimal solutions needs to be identified further. We show that optimal solutions at singular points outside the first quadrant or out of the upper bound universally allows for more prospective candidates to produce different topologies of SVMs.

The training data are labeled as $\{ X_i , y_i \} \in \mathcal{R}^d \times \{ -1, +1 \}$, $i = 1, \ldots, l$. In a typical SVM architecture, the outputs of the units established by SVs are formed by the kernel $K(X_i , X)$. This could be written as $K(X_i , X) = <\Phi(X_i), \Phi(X)>$, where $X = (x_1, \ldots, x_d)^T$, Φ is a mapping from $\mathcal{R}^d$ to high-dimensional feature space $\mathcal{H}$, $<\bullet, \bullet>$ denotes the inner product, and $(\bullet)^T$ stands for the transpose of $\bullet$. Without loss of generality, we assume that the first s vectors in the feature space are SVs.

In this paper, we study *C*-SVMs. The primal problem is

$$\min_{W,b,\xi_1,\ldots,\xi_l} \quad L_P = \|W\|^2/2 + C \sum_{i=1}^{l} \xi_i, \tag{1}$$

$$\text{s.t.} \quad 1 - \xi_i \le y_i [W^T \Phi(X_i) + b], \; i = 1, \ldots, l, \tag{2}$$

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 510–518, 2011.
© Springer-Verlag Berlin Heidelberg 2011

$$0 \le \xi_i,\ i = 1, \ldots, l, \tag{3}$$

where $0 < C$ is a constant. The Lagrangian function is

$$L = \|W\|^2/2 + C \sum_{i=1}^{l} \xi_i - \sum_{i=1}^{l} \alpha_i \{ y_i [W^T \Phi(X_i) + b] - 1 + \xi_i \} - \sum_{i=1}^{l} \lambda_i \xi_i, \tag{4}$$

where $0 \le \alpha_i$ ($i = 1, \ldots, s$). After taking differentials with respect to W and b, setting them to zero, and finally substituting the obtained equations back into L, the dual problem of (1) to (3) is built as

$$\max_{\alpha_1, \ldots, \alpha_l} \quad L_D = \sum_{i=1}^{l} \alpha_i - (1/2) \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(X_i, X_j), \tag{5}$$

$$\text{s.t.} \qquad \sum_{i=1}^{l} \alpha_i y_i = 0, \tag{6}$$

$$0 \le \alpha_i \le C,\ i = 1, \ldots, l. \tag{7}$$

The reason that people set the restriction of $0 \le \alpha_i$ ($i = 1, \ldots, s$) is that positivity of α_i ($i = 1, \ldots, s$) supports the Karush- Kuhn-Tucker (KKT) condition and the Saddle Point Theorem [1][2]. Eliminating $0 \le \alpha_i$ leads to invalidation of the derived dual programming. The research on non-positive Lagrangian multipliers is still developing [1][2]. In this paper, we first solve the dual problem with restricting $0 \le \alpha_i$, and then remove the positive requirement for α_i's.

In linear programming, negative multipliers may retain their practical significance. For example, a negative shadow price or negative Lagrangian multiplier in economics reflects greater spending resulting in lower utility [4][6][8].

SVM provides a decision function

$$f(X) = \text{sgn} [\sum_{i=1}^{s} \alpha_i^* y_i K(X_i, X) + b^*] \tag{8}$$

where sgn is the indicator function with values -1 and +1; α_i^* is the optimal Lagrange multiplier; and b^* is the optimal threshold.

Definition 1. A kernel row vector is defined by $\Lambda_i = [K(X_i, X_1), \ldots, K(X_i, X_l)] \in \mathcal{R}^{1 \times l}$, $i = 1, ..., l$.

The kernel matrix is written as

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}^- \\ \Lambda_{s+1} \\ \vdots \\ \Lambda_l \end{bmatrix} = \begin{bmatrix} \Lambda_1 \\ \vdots \\ \Lambda_s \\ \Lambda_{s+1} \\ \vdots \\ \Lambda_l \end{bmatrix} = \begin{bmatrix} K(X_1, X_1) & \cdots & K(X_1, X_l) \\ \vdots & \vdots & \vdots \\ K(X_s, X_1) & \cdots & K(X_s, X_l) \\ K(X_{s+1}, X_1) & \cdots & K(X_{s+1}, X_l) \\ \vdots & \vdots & \vdots \\ K(X_l, X_1) & \cdots & K(X_l, X_l) \end{bmatrix} \in \mathcal{R}^{s \times l}. \tag{9}$$

The remainder of the paper is organized in three sections. In Section 2, we define commonwealth points and singular points by allowing non-stationary points, as well as negative and out-of-upper-bound Lagrangian multipliers, in Lagrangian functions. We also study the geometrical structure of optimal solutions for primal and dual problems,

as well as multiple SVM architectures supported by commonwealth, including singular points. Section 3 gives two examples. Section 4 concludes the paper.

2 SVMs Supported by Commonwealth Points

The work by [3] (p. 144) has presented an approach to obtain multiple optimal solutions $\alpha_i^* + \alpha_i'$ of the dual problem by restricting α' such that (I) $0 \le \alpha_i^* + \alpha_i' \le C$, (II) $\sum_{i=1}^{s} \alpha_i' y_i = 0$, (III) $\alpha' \in \mathcal{N}(H_{ij})$ with $\mathcal{N}(\bullet)$ as the null space of $\bullet$, and (IV) $\mathbf{1}^T \alpha' = 0$ with $\mathbf{1} = (1, \ldots, 1)^T$. This is simplified by

$$(L_D^*)' = \sum_{i=1}^{l} (\alpha_i^* + \alpha_i') - (1/2) \sum_{i=1}^{l} \sum_{j=1}^{l} (\alpha_i^* + \alpha_i') \{ y_i y_j K(X_i, X_j) (\alpha_j^* + \alpha_j') \} = L_D^*. \tag{10}$$

Three weaknesses exist in this argument. First, the requirement of $0 \le \alpha_i'$ results in the only-zero solution for $\mathbf{1}^T \alpha' = 0$. Second, due to non-zero differentials, $(L_D^*)'$ is *not* a dual problem after artificially adding α',

$$\left. \frac{\partial L}{\partial W} \right|_{W=(W^*)'} = W^* - \sum_{i=1}^{l} (\alpha_i^* + \alpha_i') y_i \Phi(X_i) = -\sum_{i=1}^{l} \alpha_i' y_i \Phi(X_i) \stackrel{?}{=} 0 . \tag{11}$$

As a result, verifying the no-change of non-dual problem $(L_D^*)'$ does not disclose anything meaningful. Third, to examine the no-change in optimal solution for the primal problem, it is more important to justify the unaltered separating hyperplane. Unfortunately, $(W^*)' = \sum_{i=1}^{s} (\alpha_i^* + \alpha_i') y_i \Phi(X_i) = W^* + \sum_{i=1}^{s} \alpha_i' y_i \Phi(X_i) \ne W^*$, and $(b^*)' = 1/y_i - [(W^*)']^T \Phi(X_i) \ne b^*$.

In this paper, we remove restrictions (I) to (IV), as suggested in [3] (p. 144).

Next, we define more terms used by this paper.

Definition 2. Assume that $0 \le (\alpha_1^*, \ldots, \alpha_l^*) \le C$ is obtained from solving dual problem (5) to (7). If $((\alpha_1^*)', \ldots, (\alpha_l^*)') (\ne (\alpha_1^*, \ldots, \alpha_l^*)) \in \mathcal{R}^{1 \times l}$ preserves W^* and b^* with $(W^*)' = W^*$ and $(b^*)' = b^*$, then $(\alpha_1^*)', \ldots, (\alpha_l^*)'$ are termed as generalized Lagrangian multipliers, whereas $((\alpha_1^*)', \ldots, (\alpha_l^*)')$ is called a commonwealth point. Accordingly, the two SVM architectures with $((\alpha_1^*)', \ldots, (\alpha_l^*)')$ and $(\alpha_1^*, \ldots, \alpha_l^*)$ are named commonwealth SVMs.

The allowance of $(\alpha_i^*)' < 0$ or $C < (\alpha_i^*)'$ extends the limitation of $0 \le \alpha_i \le C$ in [3] (p. 144). Therefore, the optimal point could be located at any place in the coordinate system. In the inclusion of singular points, conditions (I) and (II) are eliminated in this paper, and we therefore have *more* solutions compared to those suggested in [3] (p. 144).

Definition 3. A Lagrangian function with generalized Lagrangian multipliers is called a generalized Lagrangian function.

A generalized Lagrangian function has a special case as the conventional Lagrangian function.

Definition 4. In the dual problem (5) to (7), if $(\alpha_1^*, \ldots, \alpha_l^*) \in \mathcal{R}^{1 \times l}$, then the dual space is called a generalized dual space.

For convenience, the objective function of the generalized dual problem is still labeled L_D^* for formality purposes.

The generalized Lagrangian function may not lead to definite programming, as it is a type of indefinite programming in SVMs. Another type of indefinite programming is a dual problem with indefinite kernels. Clearly, indefinite kernels are not generalized Lagrangian functions with generalized Lagrangian multipliers in this paper.

In [5], a rule for pruning one SV was given as follows, if s Λ_i's are linearly dependent,

Lemma 1. Assume that s Λ_i's are linearly dependent,

$$\sum_{i=1}^{s} \beta_i \Lambda_i = 0, \beta_i \in \mathcal{R}, i = 1, \ldots, s, \exists\, k, 1 \le k \le s, \beta_k \neq 0, \tag{12}$$

then the kth SV can be removed and α_i^* should be updated by

$$(\alpha_i^*)' = \alpha_i^* - (\beta_i / \beta_k)\, \alpha_k^*\, (y_k / y_i),\ i = 1, \ldots, s. \tag{13}$$

Lemma 1 can serve as a tool to relocate commonwealth point $(\alpha_1^*, \ldots, \alpha_l^*)$ to $((\alpha_1^*)', \ldots, (\alpha_l^*)')$ (see Fig. 1).

According to Definition 2, (13) is just one of the methods that can generate commonwealth points. If a nonlinear dependency among Λ_i, $i = 1, \ldots, s$, $f(\Lambda_1, \ldots, \Lambda_s) = 0$ can be found, we may also remove some SVs following a similar rule in (13). As nonlinearity incurs more complex scenarios for solutions of $f(\Lambda_1, \ldots, \Lambda_s) = 0$, we only consider the linear dependence among Λ_i's in this paper.

Theorem 1. The pruning rule (13) does not change L_P^* and L^*, $(L_P^*)' = (L^*)' = L_P^* = L^*$, but changes L_D^*, $(L_D^*)' \neq L_D^*$.

Proof:

$$\begin{aligned}
&(L_P^*)' \\
&= \|(W^*)'\|^2/2 + C \textstyle\sum_{i=1}^{l} \xi_i^* \\
&= (1/2)\, \| \textstyle\sum_{i=1, i\neq k}^{s} (\alpha_i^*)' y_i \Phi(X_i) \|^2 + C \textstyle\sum_{i=1}^{l} \xi_i^* \\
&= (1/2) \textstyle\sum_{i=1, i\neq k}^{s} [\ \alpha_i^* - (\beta_i/\beta_k)\alpha_k^* (y_k/y_i)\ ]\, y_i \Phi^T(X_i) \\
&\qquad \textstyle\sum_{j=1, j\neq k}^{s} [\ \alpha_j^* - (\beta_j/\beta_k)\alpha_k^* (y_k/y_j)\ ]\, y_j \Phi(X_j) + C \textstyle\sum_{i=1}^{l} \xi_i^* \\
&= (1/2) [\ \textstyle\sum_{i=1, i\neq k}^{s} \alpha_i^* y_i \Phi^T(X_i) + \alpha_k^* y_k \textstyle\sum_{i=1, i\neq k}^{s} (-\beta_i/\beta_k) \Phi^T(X_i)\ ] \\
&\qquad [\textstyle\sum_{j=1, j\neq k}^{s} \alpha_j^* y_j \Phi(X_j) + \alpha_k^* y_k \textstyle\sum_{j=1, j\neq k}^{s} (-\beta_j/\beta_k)\ \Phi(X_j)] + C \textstyle\sum_{i=1}^{l} \xi_i^* \\
&= (1/2) [\ \textstyle\sum_{i=1}^{s} \alpha_i^* y_i \Phi^T(X_i)\ ]\ [\ \textstyle\sum_{j=1}^{s} \alpha_j^* y_j \Phi(X_j)\ ] + C \textstyle\sum_{i=1}^{l} \xi_i^* \\
&= (1/2)\, \| \textstyle\sum_{i=1}^{s} \alpha_i^* y_i \Phi(X_i) \|^2 + C \textstyle\sum_{i=1}^{l} \xi_i^* \\
&= L_P^*.
\end{aligned} \tag{14}$$

Also,

$$\begin{aligned}
(L^*)' = \|(W^*)'\|^2/2 &+ C \textstyle\sum_{i=1}^{l} \xi_i^* - \textstyle\sum_{i=1, i\neq k}^{s} (\alpha_i^*)' \\
&\{\ y_i\ [((W^*)')^T \Phi(X_i) + (b^*)'] - 1 + \xi_i^*\ \} - \textstyle\sum_{i=1}^{l} \lambda_i^* \xi_i^*.
\end{aligned} \tag{15}$$

As

$$y_i\ [((W^*)')^T \Phi(X_i) + (b^*)'] - 1 + \xi_i^* = 0, \text{ for } 0 < (\alpha_i^*)', i = 1, \ldots, s, \tag{16}$$

and

$$\xi_i^* = 0, \text{ for } 0 < \lambda_i^*, i = 1, \ldots, l, \tag{17}$$

the same must be true for

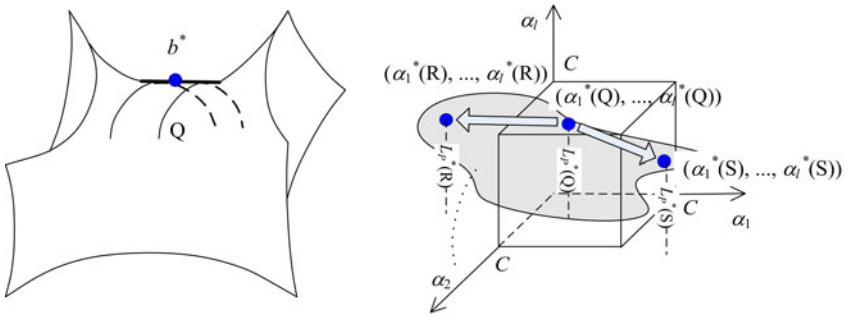
$$(L^*)' = (L_P^*)' = L_P^* = \|W^*\|^2/2 + C \sum_{i=1}^{l} \xi_i^* - \sum_{i=1}^{s} \alpha_i^* \{y_i [(W^*)^T \Phi(X_i) + b^*] - 1 + \xi_i^*\} - \sum_{i=1}^{l} \lambda_i^* \xi_i^* = L^*. \tag{18}$$

Additionally,

$$\begin{aligned} &(L_D^*)' \\ &= \sum_{i=1, i\neq k}^{s} [\, \alpha_i^* - (\beta_i/\beta_k)\alpha_k^*(y_k/y_i)\,] - (1/2) \sum_{i=1, i\neq k}^{s} \\ &\quad \sum_{j=1, j\neq k}^{s} [\, \alpha_i^* - (\beta_i/\beta_k)\alpha_k^*(y_k/y_i)\,] [\alpha_j^* - (\beta_j/\beta_k)\alpha_k^*(y_k/y_j)\,] y_i y_j K(X_i, X_j) \\ &= \sum_{i=1, i\neq k}^{s} \alpha_i^* - \sum_{i=1, i\neq k}^{s} (\beta_i/\beta_k)\alpha_k^*(y_k/y_i) - (1/2) \sum_{i=1, i\neq k}^{s} \\ &\quad \sum_{j=1, j\neq k}^{s} [\, \alpha_i^* - (\beta_i/\beta_k)\alpha_k^*(y_k/y_i)\,] [\alpha_j^* - (\beta_j/\beta_k)\alpha_k^*(y_k/y_j)\,] y_i y_j K(X_i, X_j) \\ &\neq L_D^*. \end{aligned} \tag{19}$$

As mentioned earlier, $(L_D^*)'$ in Theorem 1 is only written superficially, as general $(L_D^*)'$ is *not* a dual problem after the update of (α_i^*)'s.

Fig. 1 illustrates the geometrical structure for different scenarios of L_P^* and L^*. As optimal solution α_i^* $(i = 1, \ldots, l)$ changes, L_P^* and L^* retain the same values, $L_P^*(\mathrm{Q}) = L_P^*(\mathrm{R}) = L_P^*(\mathrm{S}) = L^*(\mathrm{Q}) = L^*(\mathrm{R}) = L^*(\mathrm{S}) = (L_P^*)'(\mathrm{Q}) = (L_P^*)'(\mathrm{R}) = (L_P^*)'(\mathrm{S}) = (L^*)'(\mathrm{Q}) = (L^*)'(\mathrm{R}) = (L^*)'(\mathrm{S})$. In Fig. 1(b), point Q represents the solution at the stationary



(a) (b)

Fig. 1. (a) Stationary point Q, and (b) geometrical structure of commonwealth points in generalized dual space. In (b), points Q, R, and S are associated with commonwealth SVMs and can be located anywhere in the coordinate system. Q denotes stationary point $(\alpha_1^*(\mathrm{Q}), \ldots, \alpha_l^*(\mathrm{Q}))$, while R and S stand for possibly non-stationary points $(\alpha_1^*(\mathrm{R}), \ldots, \alpha_l^*(\mathrm{R}))$ and $(\alpha_1^*(\mathrm{S}), \ldots, \alpha_l^*(\mathrm{S}))$, respectively. R is not in the first quadrant, and S is not in the *C*-cube. The shadow area in (b) illustrates the multiple optimal solutions in the generalized dual space corresponding to the multiple optimal solutions in the primal problem, or the dark line in (a). If only a unique solution exists for the primal problem, the dark line in (a) shrinks to a dot, while the shadow area in (b) might not generally. After finding an optimal solution for the dual problem, multiple optimal solutions can be applied, as indicated by the hollow arrows.

point. Points R and S, possibly not in the first quadrant or in the *C*-cube, denote commonwealth points, often seen at non-stationary points with non-zero differentials,

$$(\partial L/\partial W)|_{W=(W^*)'} = W^* - \sum_{i=1,\, i\neq k}^{s} [\, \alpha_i^* - (\beta_i/\beta_k)\alpha_k^* (y_k/y_i)\,]\, y_i \Phi^T(X_i)$$

$$= W^* - \sum_{i=1}^{s} \alpha_i^* y_i \Phi^T(X_i) = 0, \tag{20}$$

$$(\partial L/\partial b)|_{b=(b^*)'} = -\sum_{i=1,\, i\neq k}^{s} [\, \alpha_i^* - (\beta_i/\beta_k)\alpha_k^* (y_k/y_i)\,]\, y_i$$

$$= -\sum_{i=1,\, i\neq k}^{s} \alpha_i^* y_i + \alpha_k^* y_k \sum_{i=1,\, i\neq k}^{s} (\beta_i/\beta_k) \stackrel{?}{=} 0. \tag{21}$$

In many cases, at $((\alpha_1^*)', \ldots, (\alpha_l^*)') \in \mathcal{R}^{1\times l}$, the corresponding $(\partial L/\partial b)|_{b=(b^*)'} \neq 0$. However, setting an extra condition of $\sum_{i=1}^{s} \beta_i = 0$ enables (21) to vanish.

Theorem 2. If $\sum_{i=1}^{s} \beta_i \Lambda_i = 0$, $\beta_i \in \mathcal{R}$, $i = 1, \ldots, s$, $\exists\, k$, $1 \le k \le s$, $\beta_k \neq 0$, and $\sum_{i=1}^{s} \beta_i = 0$, then (21) vanishes.

Proof: $\sum_{i=1}^{s} \beta_i = 0$ implies $\sum_{i=1,\, i\neq k}^{s} (\beta_i/\beta_k) = -1$. It follows that (21) is zero.

As Theorem 2 does not preclude singular points, we do not elaborately evade singular points with the aid of Theorem 2.

We list Lemmas 2 and 3; the proofs can be accomplished directly by KKT [7].

Lemma 2. Assume that $(\alpha_1^*, \ldots, \alpha_l^*)$ is a solution of the dual problem. If there exists an $i \in \{1, \ldots, l\}$, such that $\alpha_i^* \in (0, C)$, then the solution of primal problem is unique for $W^* = \sum_{i=1}^{l} \alpha_i^* y_i \Phi(X_i)$ and $b^* = 1/y_j - \sum_{i=1}^{l} \alpha_i^* y_i K(X_i, X)$.

Lemma 3. Assume that $(\alpha_1^*, \ldots, \alpha_l^*)$ is a solution of the dual problem. If for all $i \in \{1, \ldots, l\}$, $\alpha_i^* = 0$, or C, then the solution of the primal problem is unique for $W^* = \sum_{i=1}^{l} \alpha_i^* y_i \Phi(X_i)$, but may not be unique for b. Specifically, $b^* \in [\, b_1, b_2\,]$ with

$$b_1 = \max \{\, \max_{j\in S-} [\, -1 - \sum_{i=1}^{l} \alpha_i^* y_i K(X_i, X_j)],$$

$$\max_{j\in V+} [\, 1 - \sum_{i=1}^{l} \alpha_i^* y_i K(X_i, X_j)]\,\}, \tag{22}$$

$$b_2 = \min \{\, \min_{j\in S+} [\, 1 - \sum_{i=1}^{l} \alpha_i^* y_i K(X_i, X_j)],$$

$$\min_{j\in V-} [\, 1 - \sum_{i=1}^{l} \alpha_i^* y_i K(X_i, X_j)]\,\}, \tag{23}$$

$$V- = \{\, i : \alpha_i^* = 0, y_i = -1\,\}, \tag{24}$$

$$V+ = \{\, i : \alpha_i^* = 0, y_i = +1\,\}, \tag{25}$$

$$S- = \{\, i : \alpha_i^* = C, y_i = -1\,\}, \tag{26}$$

$$S+ = \{\, i : \alpha_i^* = C, y_i = +1\,\}. \tag{27}$$

Corollary 1. Assume that $(\alpha_1^*, \ldots, \alpha_l^*)$ is a solution of the dual problem. If $C \to +\infty$, then the solution of primal problem is unique for both $W^* = \sum_{i=1}^{s} \alpha_i^* y_i \Phi(X_i)$ and $b^* = 1/y_j - \sum_{i=1}^{s} \alpha_i^* y_i K(X_i, X)$.

Proof: Based on $y_i ((W^*)^T X_i + b^*) = 1$, it follows that $(W^*)^T X_i + b^* = -1$ for $y_i = -1$, and $(W^*)^T X_i + b^* = +1$ for $y_i = +1$. This yields a unique solution of W^* and b^*.

Lemmas 2 and 3, and Corollary 1, show that the optimal solutions of W are always unique, yet the multiple optimal solutions of b may only occur when $C < +\infty$.

3 Examples and Discussion

We exemplify two typical scenarios.

Example 1: Let $l = 6$, $X_1 = 0$, $X_2 = 1$, $X_3 = 4$, $X_4 = 3$, $X_5 = 6$, $X_6 = 7$, $y_1 = 1$, $y_2 = 1$, $y_3 = 1$, $y_4 = -1$, $y_5 = -1$, $y_6 = -1$, and $C < +\infty$.

The primal problem is

$$\min_{W,b,\xi_1,\ldots,\xi_l} \quad L_P = \|W^*\|^2/2 + C \sum_{i=1}^{6} \xi_i^*, \tag{28}$$

$$\text{s.t.} \quad 1 - \xi_i \le y_i (W^T X_i + b), \; i = 1, \ldots, 6, \tag{29}$$

$$0 \le \xi_i, \; i = 1, \ldots, 6. \tag{30}$$

First, we find the solutions for the primal and dual problems, as well as for the generalized Lagrangian function. Herein, $L = \|W^*\|^2/2 + C \sum_{i=1}^{6} \xi_i^* - \sum_{i=1}^{6} \alpha_i [y_i (W^T X_i + b) + \xi_i - 1] - \sum_{i=1}^{6} \lambda_i \xi_i$. Let $C = 1/12$, $\rho \in (1, 4/3)$, $(\xi_1^*, \ldots, \xi_6^*) = (0, 4/3-\rho, 7/3-\rho, \rho, \rho - 1, 0)$. We obtain $W^* = -4C = -1/3$, $b^* = (1 - \xi_2) / y_2 + X_2/3 = \rho \in (1, 4/3)$, $\alpha_1^* = 0$, $\alpha_2^* = 1/12$, $\alpha_3^* = 1/12$, $\alpha_4^* = 1/12$, $\alpha_5^* = 1/12$, $\alpha_6^* = 0$, $\lambda_1^* = 1/12$, $\lambda_2^* = 0$, $\lambda_3^* = 0$, $\lambda_4^* = 0$, $\lambda_5^* = 0$, $\lambda_6^* = 1/12$. For $i = 2, 3, 4, 5$ with $0 < \alpha_i^*$, we have $b^* = (1 - \xi_2) / y_2 + X_2/3 = \rho \in [1, 4/3]$. Finally, $L_P^* = \|W^*\|^2/2 - C \sum_{i=1}^{6} \xi_i^* = 5/18$, and $L^* = L_P^* - \sum_{i=1}^{6} \alpha_i^* \{ y_i [(W^*)^T X_i + b^*] - 1 + \xi_i^* \} - \sum_{i=1}^{6} \lambda_i^* \xi_i^* = 5/18 - (1/12) [(- 1/3 + \rho - 1 + 4/3 - \rho) + (-4/3 + \rho - 1 + 7/3 - \rho) + (3/3 - \rho - 1 + \rho) + (6/3 - \rho - 1 + \rho - 1)] - \xi_1^*/12 - \xi_6^*/12 = 5/18$.

Second, in the linear dependence $\beta_2 \Lambda_2 + \beta_3 \Lambda_3 + \beta_4 \Lambda_4 + \beta_5 \Lambda_5 = 0$, we choose $\beta_3 = \alpha_3^* y_3 = 1/12$, $\beta_4 = \alpha_4^* y_4 = -1/12$, and $\beta_5 = \alpha_5^* y_5 = -1/12$. Solving for β_2 leads to $\beta_2 = 5/12$. We remove the third SV, and update the second, fourth, and fifth weights, $(\alpha_2^*)' = \alpha_2^* - (\beta_2/\beta_3) \alpha_3^* (y_3/y_2) = 1/12 - (5/12)/(1/12) (1/12) (1/1) = -1/3 < 0$, $(\alpha_4^*)' = 0$, $(\alpha_5^*)' = 0$.

Evidently, the singular point $((\alpha_1^*)', (\alpha_2^*)', (\alpha_3^*)', (\alpha_4^*)', (\alpha_5^*)', (\alpha_6^*)')$ can be described in the generalized dual space. However, this cannot be explained easily in the conventional dual space. Finally, we find that $(L_P^*)' = \|(W^*)'\|^2/2 - C \sum_{i=1}^{6} \xi_i^* = 5/18$ and $(L^*)' = (L_P^*)' - \sum_{i=2, i\neq 3}^{5} (\alpha_i^*)' \{ y_i [((W^*)')^T X_i + (b^*)'] - 1 + \xi_i^* \} - \sum_{i=1}^{6} \lambda_i^* \xi_i^* = 5/18 - (-1/3) (-1/3 + \rho - 1 + 4/3 - \rho) = 5/18 = (L_P^*)' = L_P^* = L^*$. □

In addition to a violation scenario wherein $(\alpha_i^*)'$ should be in $[0, C]$, constraint $\sum_{i=1}^{6} (\alpha_i^*)' y_i = 0$ does not hold either after pruning. Therefore, it is a singular point and we cannot establish $((\alpha_1^*)', \ldots, (\alpha_6^*)')$ as the optimal solution of the conventional dual problem. Nonetheless, these definitions and theorems can still help establish a

commonwealth SVM. The definition of commonwealth point therefore allows for the further study of more diversified optimal scenarios for different SVMs.

Example 2: Let $l = 5$, $X_1 = (0, 0)^T$, $X_2 = (-1, 0)^T$, $X_3 = (0, -1)^T$, $X_4 = (1, 0)^T$, $X_5 = (0, 1)^T$, $y_1 = -1$, $y_2 = 1$, $y_3 = 1$, $y_4 = 1$, $y_5 = 1$, and choose $C \to +\infty$.

First, we find the solutions for the primal and dual problems, as well as for the generalized Lagrangian function. The mapping from $\mathcal{R}^2$ to $\mathcal{H}$ be $\Phi(X) = (\Phi_1, \Phi_2)^T = (x_1^2, x_2^2)^T$, and $<\Phi(X_i), \Phi(X_j)> = \Phi^T(X_i)\Phi(X_j)$. A typical solution is $\alpha_1^* = 4$, $\alpha_2^* = \alpha_3^* = \alpha_4^* = \alpha_5^* = 1$. The five training data are found as SVs. Moreover, $L_P^* = 4$ and $L^* = \|W^*\|^2/2 - \sum_{i=1}^{5} \alpha_i^* \{ y_i [(W^*)^T \Phi(X_i) + b^*] - 1 \} = 4$.

Second, we let $\beta_1 = \eta \in \mathcal{R}$, $\beta_4 = \alpha_4^* y_4 = 1$, $\beta_5 = \alpha_5^* y_5 = 1$, and solve $\sum_{i=2}^{3} \beta_i \Lambda_i = -\sum_{i=1, i \neq 2, 3}^{5} \alpha_i^* y_i \Lambda_i = (0, -1, -1, -1, -1)$ for (β_2, β_3). The solution is $\beta_2 = -1$ and $\beta_3 = -1$.

As a result, while choosing the fourth SV to remove, the first and fifth SVs are pruned simultaneously, and the first to fifth weights are updated by $(\alpha_1^*)' = \alpha_1^* - (\beta_1/\beta_4)(\alpha_4^*)(y_4/y_1) = 4 - \eta = 0$ (let $\eta = 4$), $(\alpha_2^*)' = 2$, $(\alpha_3^*)' = 2$, $(\alpha_4^*)' = 0$, $(\alpha_5^*)' = 0$. Finally, $(L_P^*)' = \|(W^*)'\|^2/2 = 4 = (L^*)' = \|(W^*)'\|^2/2 - \sum_{i=2}^{3} (\alpha_i^*)' \{ y_i [((W^*)')^T \Phi(X_i) + (b^*)'] - 1 \} = 4$ is not changed. □

Solutions on the dual problem $(\alpha_1^*, \alpha_2^*, \alpha_3^*, \alpha_4^*, \alpha_5^*)$ are not unique. Nonetheless, if we let $\pi_1 = \alpha_1$, $\pi_2 = \alpha_2 + \alpha_4$, $\pi_3 = \alpha_3 + \alpha_5$, then the optimal point $(\pi_1^*, \pi_2^*, \pi_3^*) = (4, 2, 2)$ is unique. This indicates that the multiple solutions of dual problem have a certain degree of freedom.

As $\eta \in (-\infty, +\infty)$, it follows that $(\alpha_1^*)' = 4 - \eta \in (-\infty, +\infty)$. The optimal solution $((\alpha_1^*)', \ldots, (\alpha_l^*)')$ is at least a line, or there are numerous commonwealth points. Moreover, $\eta = 5$ leads to $(\alpha_1^*)' = -1$, and is located outside the first quadrant in the generalized dual space (see point R in Fig. 1).

4 Conclusions and Future Work

Multiple, sometimes, numerous, commonwealth SVMs exist in SVMs. This paper explores the geometrical structure of commonwealth points, including singular points. The values of generalized Lagrangian function and objective function for the primal problem are unaltered among the commonwealth points, while the objective function of the original dual problem shows a deviation. The exploration supports the consequence at which that the transferred SVMs with fewer SVs stand safely on non-stationary points with possibly negative or out-of-upper-bound Lagrangian multipliers, which suggests some applications with such requirements. As a byproduct, those new points may also contribute SVMs with fewer SVs.

The future work includes programming a new software package with the whole spectrum of Lagrangian multipliers as well as experimenting on larger data sets on this package.

Acknowledgments. The work was supported by the Fundamental Research Funds for the Central Universities, Research Funds of Renmin University of China (10XNI029,

Research on Financial Web Data Mining and Knowledge Management), the Natural Science Foundation of China under grant 70871001, and the 863 Project of China under grant 2007AA01Z437.

References

1. Bertsekas, D.P.: Convex Optimization Theory. Athena Scientific (2009)
2. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
3. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining Knowl. Disc. 2, 121–167 (1998)
4. Chen, R.-C., Huang, M.-R., Chung, R.-G., Hsu, C.-J.: Allocation of Short-Term Jobs to Unemployed Citizens And the Global Economic Downturn Using Genetic Algorithm. Expert Systems with Applications 38, 7535–7543 (2011)
5. Liang, X., Chen, R., Guo, X.: Pruning Support Vector Machines Without Altering Performances. IEEE Trans. Neural Netw. 19, 1792–1803 (2008)
6. Mankiw, N.G.: Mcroeconomics, 6th edn. Worth Publishers (2006)
7. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience (1998)
8. Varian, H.R.: Intermediate Microeconomics: A Modern Approach, 7th edn. Norton and Company (2005)

Learning with Box Kernels

Stefano Melacci and Marco Gori

Department of Information Engineering
University of Siena, 53100 Siena, Italy
{mela,marco}@dii.unisi.it

Abstract. Supervised examples and prior knowledge expressed by propositions have been profitably integrated in kernel machines so as to improve the performance of classifiers in different real-world contexts. In this paper, using arguments from variational calculus, a novel representer theorem is proposed which solves optimally a more general form of the associated regularization problem. In particular, it is shown that the solution is based on *box kernels*, which arises from combining classic kernels with the constraints expressed in terms of propositions. The effectiveness of this new representation is evaluated on real-world problems of medical diagnosis and image categorization.

Keywords: Box kernels, Constrained variational calculus, Kernel machines, Propositional rules.

1 Introduction

The classic supervised learning framework is based on a collection of ℓ labeled points, $\mathcal{L} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, \ell\}$, where $\boldsymbol{x}_i \in \mathcal{X} \subset I\!R^d$ and $y_i \in \{-1, 1\}$. This paper focuses on supervised learning from $\ell_{\mathcal{X}}$ labeled regions of the input space, $\mathcal{L}_{\mathcal{X}} = \{(\mathcal{X}_j, y_j),\ j = 1, \ldots, \ell_{\mathcal{X}}\}$, where $\mathcal{X}_j \in 2^{\mathcal{X}}$, and $y_j \in \{-1, 1\}$. Of course, these regions can degenerate to single points and it is convenient to think of the available supervision without distinguishing between the supervised entities, so as one considers to deal with $\ell_t := \ell + \ell_{\mathcal{X}}$ labeled pairs. The case of multi dimensional intervals,

$$\mathcal{X}_j = \{\boldsymbol{x} \in I\!R^d :\ x^z \in [a_j^z, b_j^z],\ z = 1, \ldots, d\}, \tag{1}$$

where $\boldsymbol{a}_j, \boldsymbol{b}_j \in I\!R^d$ collect the lower and upper bounds, respectively, is the one which is more relevant in practice. The pair $(\mathcal{X}_j, y_j)$ formalizes the knowledge that a supervisor provides in terms of

$$\forall \boldsymbol{x} \in I\!R^d,\quad \bigwedge_{z=1}^{d} \left((x^z \geq a_j^z) \wedge (x^z \leq b_j^z)\right) \Rightarrow\ class(y_j), \tag{2}$$

so as we can interchangeably refer to it as *labeled box region* or *propositional rule*[1]. This framework has been introduced in a number of papers and its potential impact in real-world applications has been analyzed in different contexts (see

[1] While this can be thought of as FOL formula, it is easy to see that the quantifier is absorbed in the involved variables and that we simply play with propositions.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 519–528, 2011.
© Springer-Verlag Berlin Heidelberg 2011

e.g. [1] and references therein). Most of the research in this field can be traced back to Fung et al. (2002) [2] who proposed to embed labeled (polyhedral) sets into Support Vector Machines (SVMs); the corresponding model was referred to as Knowledge-based SVM (KSVM) and it has been the subject of a number of significant related studies [3,4,5,6].

This paper proposes an in-depth revision of those studies that is inspired by the approach to regularization networks of [7]. The problem of learning is properly re-formulated by the natural expression of supervision on sets, which results in the introduction of a loss function that fully involves them. Basically, any set $\mathcal{X}_j$ is associated with the characteristic function $c_{\mathcal{X}_j}(\boldsymbol{x})$, and its normalized form $\hat{c}_{\mathcal{X}_j}(\boldsymbol{x}) := c_{\mathcal{X}_j}(\boldsymbol{x}) / \int_{\mathcal{X}} c_{\mathcal{X}_j}(\boldsymbol{x}) d\boldsymbol{x}$ degenerates to the Dirac distribution $\delta(\boldsymbol{x} - \boldsymbol{x}_j)$ in the case in which $\mathcal{X}_j = \{\boldsymbol{x}_j\}$. Interestingly, it is shown that the solution emerging from the regularized learning problem does not lead to the kernel expansion on the available data points, and the kernel is no longer the Green's function of the associated regularization operator (see [8] and [9], pag. 94). A new representer theorem is given, which indicates an expansion into two different kernels. The first one corresponds with the Green's function of the stabilizer, while the second one is a *box kernel*. Basically, the box kernel is the outcome of the chosen regularization operator and of the structure of the box region. When the region degenerates to a single point, the two kernel functions perfectly match. This goes beyond the discretization of the knowledge sets, that would make the problem rapidly intractable when the dimensionality of the input space increases. In addition, we provide an explicit expression of box kernels in the case of the regularization operator associated to the Gaussian kernel, but the proposed framework suggests extensions to other cases. The analysis clearly shows why the explicit expression becomes easy in the case of boxes, whereas for general sets this seems to be hard. However, most interesting applications suggests a scenario in which logic statements in form of propositions help, which reduces sets to boxes. The experiments indicate that the proposed approach achieves state of the art results with clear improvements in some cases.

2 Learning from Labeled Sets

We formulate the problem of learning from labeled sets and/or labeled points in a unique framework, simply considering that each point corresponds to a singleton. More formally, given a labeled set $\mathcal{X}_j$, the characteristic function $c_{\mathcal{X}_j}(\boldsymbol{x})$ associated with it is 1 when $\boldsymbol{x} \in \mathcal{X}_j$, otherwise it is 0. If $vol(\mathcal{X}_j)$ is the measure of the set, $vol(\mathcal{X}_j) = \int_{\mathcal{X}} c_{\mathcal{X}_j}(\boldsymbol{x}) d\boldsymbol{x}$, the normalized characteristic function is $\hat{c}_{\mathcal{X}_j}(\boldsymbol{x}) := c_{\mathcal{X}_j}(\boldsymbol{x}) / vol(\mathcal{X}_j)$, and when the set degenerates to a single point $\boldsymbol{x}_j$, then $\hat{c}_{\mathcal{X}_j}(\boldsymbol{x})$ is the Dirac delta $\delta(\boldsymbol{x} - \boldsymbol{x}_j)$.

Following the popular framework for regularized function learning [7] we seek for a function f belonging to $\mathcal{F} = W^{k,p}$, the subset of L^p whose functions admit derivatives up to some order k. We introduce the term $m_{\mathcal{X}_j}(f) := \int_{\mathcal{X}} f(\boldsymbol{x}) \hat{c}_{\mathcal{X}_j}(\boldsymbol{x}) d\boldsymbol{x}$, that is the average value of f over $\mathcal{X}_j$. Of course, when $\mathcal{X}_j =$

$\{\boldsymbol{x}_j\}$ we get $m_{\mathcal{X}_j}(f) = f(\boldsymbol{x}_j)$. The problem of learning from labeled sets can be formulated as the minimization of

$$R_m[f] := \sum_{h \in \mathbb{N}_{\ell_t}} V(y_h, m_{\mathcal{X}_h}(f)) + \lambda \|Pf\|^2, \tag{3}$$

where $V \in C^1(\{-1,1\} \times \mathbb{R}, \mathbb{R}^+)$ is a convex loss function, $\mathbb{N}_m$ denotes the set of the first m integers, and $\lambda > 0$ weights the effect of the regularization term. P is a pseudo-differential operator, which admits the adjoint $P^\star$, so that $\|Pf\|^2 =< Pf, Pf >=< f, P^\star Pf >=< f, Lf >$, being $L = P^\star P$. The unconstrained formulation of (3) allows the classifier to handle noisy supervisions, as required in real-world applications. Moreover, a positive scalar value can be associated to each term of the sum to differently weight the contribute of each labeled element.

Theorem 1. *Let $KerL = \{0\}$ be, and let g be the Green's function of L. Then $R_m[\cdot]$ admits the unique minimum*

$$f^\star = \sum_{j \in \mathbb{N}_{\ell_t}} \alpha_j \beta(\mathcal{X}_j, \boldsymbol{x}), \tag{4}$$

where $\beta(\mathcal{X}_j, \boldsymbol{x}) := \int_{\mathcal{X}} g(\boldsymbol{x}, \boldsymbol{\varsigma}) \hat{c}_{\mathcal{X}_j}(\boldsymbol{\varsigma}) d\boldsymbol{\varsigma}$, and α_j are scalar values.

Proof: Any weak extreme of $R_m[f]$ satisfies the Euler-Lagrange equation

$$Lf(\boldsymbol{x}) = -\frac{1}{\lambda} \sum_{j \in \mathbb{N}_{\ell_t}} V_f'\left(y_j, m_{\mathcal{X}_h}(f)\right) \cdot \hat{c}_{\mathcal{X}_j}(\boldsymbol{x}) \tag{5}$$

where $V_f' = \frac{\partial V}{\partial f}$. This comes straightforwardly from variational calculus ([10], pag. 16). Since $KerL = \{0\}$ the functional $< f, Lf >$ is strictly convex which, considering that $V(y_h, \cdot)$ is also convex, leads us to conclude that any extreme of $R_m[\cdot]$ collapses to the unique minimum $f^\star$. Now, $\forall \boldsymbol{x} \in \mathcal{X}$ let $g(\boldsymbol{x}, \cdot) : \; Lg(\boldsymbol{x}, \boldsymbol{\varsigma}) = \delta(\boldsymbol{x} - \boldsymbol{\varsigma})$ be the Green's function of L. When using again the hypothesis $KerL = \{0\}$ we can invert L, from which the thesis follows. ∎

If we separate the contributions coming from points and sets the above representer theorem can be re-written as $f^\star(\boldsymbol{x}) = \sum_{i \in \mathbb{N}_\ell} \alpha_i g(\boldsymbol{x}_i, \boldsymbol{x}) + \sum_{j \in \mathbb{N}_{\ell_\mathcal{X}}} \alpha_j \beta(\mathcal{X}_j, \boldsymbol{x})$.

Now, let us define

$$\mathcal{K}(\mathcal{X}_i, \mathcal{X}_j) := \int_{\mathcal{X}} \beta(\mathcal{X}_i, \boldsymbol{x}) \cdot \hat{c}_{\mathcal{X}_j}(\boldsymbol{x}) d\boldsymbol{x}. \tag{6}$$

The following proposition gives insights on the cases in which either $\mathcal{X}_i$ or $\mathcal{X}_j$ degenerate to points.

Proposition 1.

$$i.\ \mathcal{K}(\mathcal{X}_i, \{\boldsymbol{x}_j\}) = \beta(\mathcal{X}_i, \boldsymbol{x}_j) \tag{7}$$

$$ii.\ \mathcal{K}(\{\boldsymbol{x}_i\}, \{\boldsymbol{x}_j\}) = g(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{8}$$

PROOF:

i. If $\mathcal{X}_j = \{\boldsymbol{x}_j\}$ then $\hat{c}_{\mathcal{X}_j}(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_j)$ and the thesis follows from 6.
ii. If in addition to the above hypothesis we also have $\mathcal{X}_i = \{\boldsymbol{x}_i\}$ then $\hat{c}_{\mathcal{X}_i}(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_i)$, again, yields the thesis when invoking 6. ■

The function $\mathcal{K}$ makes it possible to devise an efficient algorithmic scheme based on the collapsing to a finite dimension of the infinite dimensional optimization problem of finding weak minima for $R_m[\cdot]$ (3). Now we formally prove this aspect.

Theorem 2. *When the hypotheses of Theorem 1 hold true then* $R_m[f^\star]{=}\mathcal{R}_m(\boldsymbol{\alpha})$, *where*

$$\mathcal{R}_m(\boldsymbol{\alpha}) = \sum_{i \in I\!N_{\ell_t}} V(y_i, \sum_{j \in I\!N_{\ell_t}} \alpha_j \mathcal{K}(\mathcal{X}_j, \mathcal{X}_i)) + \lambda \sum_{i,j \in I\!N_{\ell_t}} \alpha_i \alpha_j \mathcal{K}(\mathcal{X}_i, \mathcal{X}_j). \quad (9)$$

PROOF: When plugging $f^\star$ expressed by 4 into 3 and using $Lg = \delta$, we get

$$\begin{aligned} R_m[f^\star] &= \sum_{i \in I\!N_{\ell_t}} V(y_i, \int_{\mathcal{X}} \sum_{j \in I\!N_{\ell_t}} \alpha_j \beta(\mathcal{X}_j, \boldsymbol{x}) \hat{c}_{\mathcal{X}_i}(\boldsymbol{x}) d\boldsymbol{x}) \\ &\quad + \lambda < \sum_{i \in I\!N_{\ell_t}} \alpha_i \beta(\mathcal{X}_i, \boldsymbol{x}), L(\sum_{i \in I\!N_{\ell_t}} \alpha_j \beta(\mathcal{X}_j, \boldsymbol{x})) > \\ &= \sum_{i \in I\!N_{\ell_t}} V(y_i, \sum_{j \in I\!N_{\ell_t}} \alpha_j \int_{\mathcal{X}} \beta(\mathcal{X}_j, \boldsymbol{x}) \hat{c}_{\mathcal{X}_i}(\boldsymbol{x}) d\boldsymbol{x}) \\ &\quad + \lambda \sum_{i,j \in I\!N_{\ell_t}} \alpha_i \alpha_j < \beta(\mathcal{X}_i, \boldsymbol{x}), \hat{c}_{\mathcal{X}_j}(\boldsymbol{x}) > \end{aligned} \quad (10)$$

and the thesis follows when applying definition 6. ■

3 Box Kernels

The function $\mathcal{K}(\cdot,\cdot)$ comes out from the kernel $g(\cdot,\cdot)$ and returns a number which depends on its operands that can be space regions or points. Now we consider regions bounded by multi dimensional intervals (boxes), so that $\mathcal{K}(\cdot,\cdot)$ is referred to as the *box kernel* coming from g. These regions formalizes the type of knowledge that we introduced in Section 1, and $vol(\mathcal{X}_j) = \prod_{i=1}^{d} |a_j^i - b_j^i|$. The box kernel can be plugged in every existing kernel based classifier, allowing it to process labeled box regions without any modification to the learning algorithm. The function $\mathcal{K}(\cdot,\cdot)$ inherits a number of properties from the kernel $g(\cdot,\cdot)$.

Proposition 2. *Let* $I\!K \in I\!R^{\ell_t,\ell_t}$ *be the Gram matrix associated with the function* $\mathcal{K}(\mathcal{X}_i, \mathcal{X}_j)$. *If* g *is a positive definite kernel function then* $I\!K \geq 0$.

PROOF: We distinguish three cases:

i. $vol(\mathcal{X}_i),\ vol(\mathcal{X}_j) > 0$. Since $g > 0$ there exists $\phi : \ \forall \boldsymbol{x}, \boldsymbol{\varsigma} \in \mathcal{X} : \ g(\boldsymbol{x}, \boldsymbol{\varsigma}) = < \phi(\boldsymbol{x}), \phi(\boldsymbol{\varsigma}) >$. From the definition 6 we get

$$\begin{aligned}\mathcal{K}(\mathcal{X}_i, \mathcal{X}_j) &= \int_{\mathcal{X}} (\int_{\mathcal{X}} g(\boldsymbol{x}, \boldsymbol{\varsigma}) \hat{c}_{\mathcal{X}_i}(\boldsymbol{\varsigma}) d\boldsymbol{\varsigma}) \hat{c}_{\mathcal{X}_j}(\boldsymbol{x}) d\boldsymbol{x} \\ &= \int_{\mathcal{X}} \int_{\mathcal{X}} < \phi(\boldsymbol{x}), \phi(\boldsymbol{\varsigma}) > \hat{c}_{\mathcal{X}_i}(\boldsymbol{\varsigma}) \hat{c}_{\mathcal{X}_j}(\boldsymbol{x}) d\boldsymbol{\varsigma} d\boldsymbol{x} \\ &= < \int_{\mathcal{X}} \phi(\boldsymbol{x}) \hat{c}_{\mathcal{X}_j}(\boldsymbol{x}) d\boldsymbol{x}, \int_{\mathcal{X}} \phi(\boldsymbol{\varsigma}) \hat{c}_{\mathcal{X}_i}(\boldsymbol{\varsigma}) d\boldsymbol{\varsigma} > = < \Phi(\mathcal{X}_i), \Phi(\mathcal{X}_j) > \end{aligned} \quad (11)$$

where $\Phi(\mathcal{Z}) := \int_{\mathcal{Z}} \phi(\boldsymbol{x}) \hat{c}_{\mathcal{Z}}(\boldsymbol{x}) d\boldsymbol{x}$ being $\mathcal{Z} \in 2^{\mathcal{X}}$.

ii. $vol(\mathcal{X}_i) > 0$ and $\mathcal{X}_j = \{\boldsymbol{x}_j\}$. Following the same arguments as above,

$$\begin{aligned}\mathcal{K}(\mathcal{X}_i, \{\boldsymbol{x}_j\}) &= \int_{\mathcal{X}} (\int_{\mathcal{X}} g(\boldsymbol{x}, \boldsymbol{\varsigma}) \hat{c}_{\mathcal{X}_i}(\boldsymbol{\varsigma}) d\boldsymbol{\varsigma}) \delta(\boldsymbol{x} - \boldsymbol{x}_j) d\boldsymbol{x} \\ &= \int_{\mathcal{X}} g(\boldsymbol{x}_j, \boldsymbol{\varsigma}) \hat{c}_{\mathcal{X}_i}(\boldsymbol{\varsigma}) d\boldsymbol{\varsigma} \\ &= < \phi(\boldsymbol{x}_j), \int_{\mathcal{X}} \phi(\boldsymbol{\varsigma}) \hat{c}_{\mathcal{X}_i}(\boldsymbol{\varsigma}) d\boldsymbol{\varsigma} > = < \phi(\boldsymbol{x}_j), \Phi(\mathcal{X}_i) > \end{aligned} \quad (12)$$

and $\phi(\boldsymbol{x}_z)$ is the degenerate case of $\Phi(\mathcal{Z})$, in which $\mathcal{Z}$ becomes a point $\boldsymbol{x}_z$.

iii. $\mathcal{X}_i = \{\boldsymbol{x}_i\}$ and $\mathcal{X}_j = \{\boldsymbol{x}_j\}$. In this case we immediately get $\mathcal{K}(\mathcal{X}_i, \mathcal{X}_j) = g(\boldsymbol{x}_i, \boldsymbol{x}_j) = < \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) >$.

Finally, if we construct the Gram matrix $\mathbb{K}$ using *i*, *ii*, and *iii* the thesis comes out straightforwardly. ■

Gaussian Kernels. Now and in the rest of the paper, we focus attention on the case in which g is Gaussian kernel of width σ, $g(\boldsymbol{x}, \boldsymbol{z}) = \exp(-0.5 \left\| \boldsymbol{x} - \boldsymbol{z} \right\|^2 \sigma^{-2})$. However our framework is generic, and the extension to other cases follows similar analyses.

Proposition 3. *If g is a Gaussian kernel then*

$$\beta(\mathcal{X}_j, \boldsymbol{x}) = \frac{1}{vol(\mathcal{X}_j)} \prod_{i=1}^{d} \frac{(\sqrt{2\pi}\sigma)}{2} (erfc(\frac{x^i - b_j^i}{\sqrt{2}\sigma}) - erfc(\frac{x^i - a_j^i}{\sqrt{2}\sigma})) \quad (13)$$

PROOF: We recall that the isotropic Gaussian kernel is given by the products of d Gaussian kernels that independently operate in each dimension. Since $\mathcal{X}_j$ is a box region, we can rewrite the integral over $\mathcal{X}_j$ into a product of d definite integrals. In detail,

$$\beta(\mathcal{X}_j, \boldsymbol{x}) \cdot vol(\mathcal{X}_j) = \int_{\mathcal{X}_j} e^{\frac{\|\boldsymbol{x} - \boldsymbol{\zeta}\|^2}{-2\sigma^2}} d\boldsymbol{\zeta} \ = \ \prod_{i=1}^{d} \int_{a_j^i}^{b_j^i} e^{\frac{(x^i - \zeta^i)^2}{-2\sigma^2}} d\zeta^i$$

$$= \prod_{i=1}^{d} \left(\int_{a_j^i}^{+\infty} e^{\frac{(x^i-\zeta^i)^2}{-2\sigma^2}} d\zeta^i - \int_{b_j^i}^{+\infty} e^{\frac{(x^i-\zeta^i)^2}{-2\sigma^2}} d\zeta^i \right)$$

$$= \prod_{i=1}^{d} \frac{(\sqrt{2\pi}\sigma)}{2} \left(erfc\left(\frac{x^i - b_j^i}{\sqrt{2}\sigma}\right) - erfc\left(\frac{x^i - a_j^i}{\sqrt{2}\sigma}\right) \right) \quad (14)$$

where $erfc(x) = \frac{2}{\sqrt{\pi}} \int_z^{+\infty} e^{-t^2} dt$ is the complementary error function. ■

Proposition 4. *If g is a Gaussian kernel then*

$$\mathcal{K}(\mathcal{X}_h, \mathcal{X}_k) = q_{h,k} \cdot \prod_{i=1}^{d} \left(\Psi(b_h^i, b_k^i) - \Psi(a_h^i, b_k^i) \ -\Psi(b_h^i, a_k^i) + \Psi(a_h^i, a_k^i) \right) \quad (15)$$

where

$$\Psi(a, b) := \frac{a-b}{\sqrt{2}\sigma} erfc\left(\frac{a-b}{\sqrt{2}\sigma}\right) - \frac{1}{\sqrt{\pi}} e^{-\frac{(a-b)^2}{2\sigma^2}}, \quad (16)$$

$$q_{h,k} := \frac{(\sqrt{\pi}\sigma^2)^d}{vol(\mathcal{X}_\kappa) vol(\mathcal{X}_h)}. \quad (17)$$

Proof: Given $p_{h,k} := \frac{(\sqrt{2\pi}\sigma)^d}{2^d vol(\mathcal{X}_\kappa) vol(\mathcal{X}_h)}$ we have

$$\mathcal{K}(\mathcal{X}_h, \mathcal{X}_k) := \int_{\mathcal{X}_k} \frac{\beta(\mathcal{X}_h, \boldsymbol{x})}{vol(\mathcal{X}_\kappa)} d\boldsymbol{x} = p_{h,k} \int_{\mathcal{X}_k} \prod_{i=1}^{d} \left(erfc\left(\frac{x^i - b_h^i}{\sqrt{2}\sigma}\right) - erfc\left(\frac{x^i - a_h^i}{\sqrt{2}\sigma}\right) \right) d\boldsymbol{x}$$

$$= p_{h,k} \prod_{i=1}^{d} \left(\int_{a_k^i}^{b_k^i} erfc\left(\frac{x^i - b_h^i}{\sqrt{2}\sigma}\right) dx^i - \int_{a_k^i}^{b_k^i} erfc\left(\frac{x^i - a_h^i}{\sqrt{2}\sigma}\right) dx^i \right) \quad (18)$$

that must be paired with $\int erfc(z)dz = z \cdot erfc(z) - e^{-z^2}(\sqrt{\pi})^{-1}$ to complete the proof. ■

In Fig. 1 we report an illustrative example of $\mathcal{K}(\mathcal{X}_i, \mathcal{X}_j)$ where $\mathcal{X}_j = \{\boldsymbol{x}_j\}$ and $\mathcal{X}_i$ is progressively reduced until it degenerates to a point, leading to the classical Gaussian kernel. Using a synthetic data set, Fig. 2 (a-c) show the separation hyperplane of a box-kernel-based SVM, trained with labeled points, labeled box regions, or both of them, respectively. The optimal separation boundary between the two classes becomes nonlinear when introducing the labeled regions, and it is correctly modeled by the box kernel. Fig. 2 (d) considers the effect of increasing the parameter λ, and it shows how a soft margin estimate is allowed within the available box regions, increasing the robustness to noisy supervisions.

In Fig. 2 (e) not all the training points are coherent with the knowledge sets. The averaging effect of the box kernel within each labeled box region, introduced in (3) by the $m_{\mathcal{X}_j}(f)$ term, allows the classifier to handle this situation. As a matter of fact, SVMs exploits a hinge loss for the labeled entities, and the (absolute) max value of f is larger inside the region in which we find the incoherency, so that its average still matches the corresponding box label (Fig. 2 (f)). The regularized nature of the learning problem does not allow the value of f to explode to infinity.

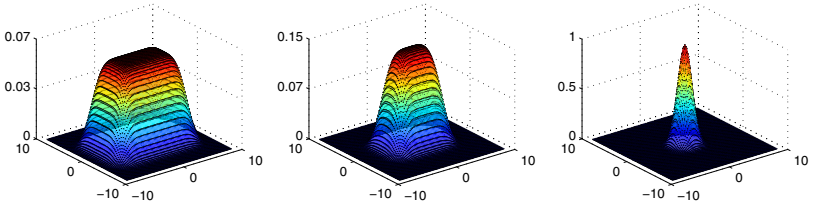


Fig. 1. The $\mathcal{K}(\mathcal{X}_i, \mathcal{X}_j)$ function (g is Gaussian) where $\mathcal{X}_j = \{\boldsymbol{x}_j\}$ and $\mathcal{X}_i$ is defined from $[-6, -4]$ to $[6, 4]$ and it is progressively reduced until it degenerates to a point (left to right). The last picture corresponds to a Gaussian kernel.

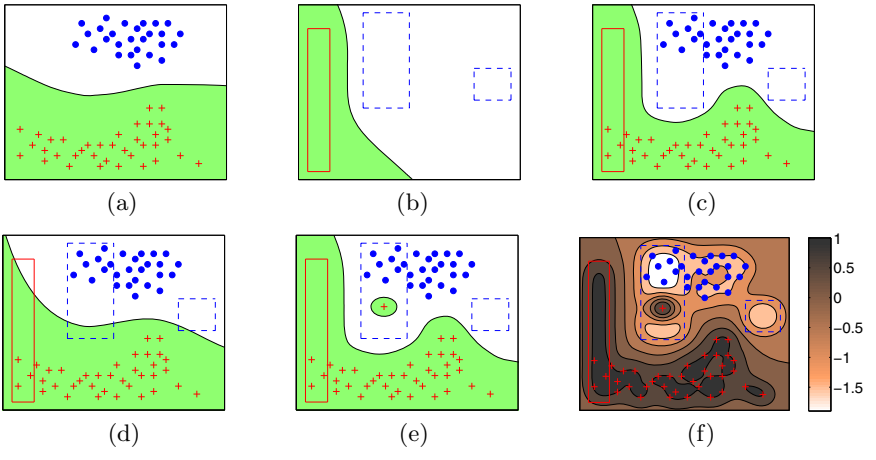


Fig. 2. SVM trained on a 2-class dataset using the box kernel (red crosses/boxes: class +1, blue circles/boxes: class -1). (a) The separation boundary when only labeled points are used; (b) using labeled box regions only; (d) using both labeled points and regions; (e) using a larger λ (it penalizes the data fitting); (e) a labeled point (+) is incoherent with the leftmost blue-dotted box; (f) the level curves of f in the case of (e).

4 Experimental Results

We ran comparative experiments that are based on real-world scenarios: diagnosing diabetes, and recognizing handwritten digits. Before going into further details, we shortly describe the related algorithms.

In [2] the authors formalize a constrained linear optimization problem based on the available rules (i.e. labeled regions), that leads to a linear classification function (KSVM, Knowledge-based SVM). The extension of the KSVM framework to the nonlinear case has been studied in [3]. However, the nonlinear "kernelization" is not a transparent procedure that can be easily related to the original knowledge, making the approach less practical. Le et. al [4] proposed a simpler alternative, that we will refer to as SKSVM (Simpler KSVM). An SVM is trained from labeled points only, excluding the ones that fall in the (arbitrary

shaped) labeled regions, and, at test time, its prediction is post processed to match the available knowledge. The main drawback of this approach is that it is not able to generalize from knowledge on labeled regions only. A more recent idea was proposed in [5,6]. A kernel-based classifier is extended to model labeled nonlinear space regions by the discretization of the supervised space on a preselected subset of points. This criterion was applied to a linear programming SVM [5] (NKC, Nonlinear Knowledge-based Classifier) and to a proximal nonlinear classifier [6] (PKC, Proximal Knowledge-based Classifier). However, it is unclear how to sample the regions on which prior knowledge is given, and a considerable amount of points may be needed, especially in high dimensions.

In each experiment, the features that are not involved in the available rules are bounded by their *min*, *max* values over the entire data collection. Classifier parameters were chosen by ranging them over dense grid of values in $[10^{-5}, 10^{5}]$, and using a cross-validation procedure (described below).

Diabetes. The Pima Indian Diabets [11] dataset is composed by the results of 8 medical tests for 768 female patients at least 21 years old of Pima Indian heritage. The task is to predict whether the patient shows signs of diabetes. KSVMs have been recently evaluated in this data [12], and we replicated the same experimental setting. Two rules from the National Institute of Health are defined, involving the second (PLASMA) and sixth (MASS) features,

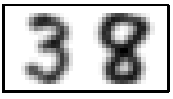
$$(MASS \geq 30) \wedge (PLASMA \geq 126) \Rightarrow positive$$
$$(MASS \leq 25) \wedge (PLASMA \leq 100) \Rightarrow negative.$$

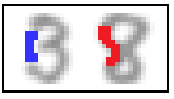
We note that the rules can be applied to directly classify 269 instances, and only 205 of them will be *correctly* classified. A collection of 200 random points is used to train the classifiers, 30 points to validate their parameters, whereas the results of Table 1 are computed on the rest of the data, averaged over 20 runs. When using rules and labeled point, BOX shows a slightly better accuracy than KSVM but the two results are essentially equivalent. We noted that the information carried in the labeled data points is enough to fulfill the box constraints. Differently, when only rules (i.e. labeled box regions) are fed to the classifier, a nonlinear estimate resulted more appropriate, and BOX shows a significant improvement with respect to KSVM.

Handwritten Digit Recognition. The USPST is the test collection of 16x16 pictures of 2007 handwritten digits from the US Postal System. We consider

Table 1. The average accuracy and standard deviation on the Diabetes data in the setup of [12] (KSVM)

Method	Mean Accuracy	Std
KSVM (rules only)	64.23%	1.19%
BOX (rules only)	**70.44%**	1.03%
KSVM	76.33%	0.63%
BOX	**76.39%**	1.30%





$$(Intensity\ of\ the\ blue\ region\ in\ (b) \geq 220) \Rightarrow 3$$
$$(Intensity\ of\ the\ red\ region\ in\ (b) \leq 160) \Rightarrow 8$$

(a) (b) (c)

Fig. 3. (a) Examples of digits 3 and 8 from USPST; (b) the region in which additional knowledge is provided to distinguish between the two classes (18 blue pixels in for class 3 and and 24 red pixels for class 8); (c) the rules provided for this task

the task of predicting whether an input image, represented as a vector of gray scale intensities, is a 3 or a 8. Their representations are often very similar, and when the number of labeled training points is small the classification task is challenging. Given the pair of examples of Fig. 3 (a), a volunteer indicated the portions of image that he considered more useful to distinguish them (Fig. 3 (b)). He also provided the ranges of intensity values that he would tolerate in each region, considering that not all the data will perfectly match the given pair. The resulting rules are reported in Fig. 3 (c). We randomly generated training/validation and test splits, repeating the process 20 times. The former group was composed of 10 labeled points only (4 of them were used to validate the classifier parameters). The pair of Fig. 3 was included in all the training sets. We compared all the described algorithms, collecting the results in Table 2. A Gaussian kernel was used for nonlinear classifiers. BOX compares favorably with all the other methods, also when only the box rules are provided to the classifier. This result is remarkable, since the rules only applies to 46 out of 338 data points. Differently, SKSVM suffers from the removal of the training examples that fulfill the given rules, whereas in KSVM is hard to find a good trade-off between rule fulfillment and labeled points matching. NKC and PKC require a discrete sampling of the labeled region, so that we provided those algorithms with 100 additional training points, generated by adding random noise to the pair of Fig. 3. However, this process is very heuristic, and BOX resulted in better accuracy without the need of any discrete sampling.

Table 2. The average accuracy and standard deviation of 20 experiments on USPST 3vs8 for different algorithms

Method	Mean Accuracy	Std
KSVM (rules only)	79.42%	0.28%
NKC/PKC (rules only)	77.38%	0.35%
BOX (rules only)	**80.72%**	0.35%
SVM	89.78%	5.35%
SKSVM	87.87%	5.03%
KSVM	89.57%	5.70%
NKC/PKC	90.72%	4.46%
BOX	**92.55%**	4.43%

5 Conclusions

Based on the inspiring framework given by [7], in this paper, we give a unified variational framework of the class of problems introduced in [2], that incorporate both supervised points and supervised sets and prove a new representer theorem for the optimal solution. It turns out that the solution is based on a novel class of kernels, referred to as *box kernels*, that are created by joining a classic kernel with the collection of supervised sets - that can degenerate to points. Interestingly, supervised points and sets are treated differently by box kernels, since they adapt their shape to the measure of the sets. This suggestion of the box kernel for the problem at hand, which derives from the more general variational formulation, is the most distinguishing feature of the proposed approach. Interestingly, the algorithmic issues that hold for kernel machines still apply, which makes it easy the actual experimentation of the approach. The given set of experiments show that the proposed solution is equivalent to or compares favorably with the state-of-the art in this field, overcoming several issues of the related algorithms. Finally, it is worth mentioning that the proposed approach of carving new kernels from the specific problem might open the doors to other solutions for different forms of prior knowledge.

References

1. Lauer, F., Bloch, G.: Incorporating prior knowledge in support vector machines for classification: a review. Neurocomputing 71(7-9), 1578–1594 (2008)
2. Fung, G., Mangasarian, O., Shavlik, J.: Knowledge-based support vector machine classifiers. In: Advances in NIPS, pp. 537–544 (2002)
3. Fung, G.M., Mangasarian, O.L., Shavlik, J.: Knowledge-Based Nonlinear Kernel Classifiers. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 102–113. Springer, Heidelberg (2003)
4. Le, Q., Smola, A., Gártner, T.: Simpler knowledge-based support vector machines. In: Proceedings of ICML, pp. 521–528. ACM (2006)
5. Mangasarian, O., Wild, E.: Nonlinear knowledge-based classification. IEEE Trans. on Neural Networks 19(10), 1826–1832 (2008)
6. Mangasarian, O., Wild, E., Fung, G.: Proximal Knowledge-based Classification. Statistical Analysis and Data Mining 1(4), 215–222 (2009)
7. Poggio, T., Girosi, F.: A theory of networks for approximation and learning. Technical report. MIT (1989)
8. Schoelkopf, B., Smola, A.: From regularization operators to support vector kernels. In: M. Kaufmann (ed.) Advances in NIPS (1998)
9. Schoelkopf, B., Smola, A.: Learning with kernels. The MIT Press (2002)
10. Giaquinta, M., Hildebrand, S.: Calculus of Variations I, vol. 1. Springer, Heidelberg (1996)
11. Frank, A., Asuncion, A.: UCI repository (2010)
12. Kunapuli, G., Bennett, K., Shabbeer, A., Maclin, R., Shavlik, J.: Online Knowledge-Based Support Vector Machines. In: ECML, pp. 145–161 (2010)

A Novel Parameter Refinement Approach to One Class Support Vector Machine

Trung Le, Dat Tran, Wanli Ma, and Dharmendra Sharma

Faculty of Information Sciences and Engineering
University of Canberra, ACT 2601, Australia
{trung.le,dat.tran,wanli.ma,dharmendra.sharma}@canberra.edu.au

Abstract. One-Class Support Vector Machine employs a grid parameter selection process to discover the best parameters for a given data set. It is assumed that two separate trade-off parameters are assigned to normal and abnormal data samples, respectively. However, this assumption is not always true because data samples have different contributions to the construction of hypersphere or hyperplane decision boundary. In this paper, we introduce a new iterative learning process that is carried out right after the grid parameter selection process to refine the trade-off parameter value for each sample. In this learning process, a weight is assigned to each sample to represent the contribution of that sample and is iteratively refined. Experimental results performed on a number of data sets show a better performance for the proposed approach.

Keywords: One class classification, novelty detection, support vector machine, machine learning.

1 Introduction

In one-class classification, we mainly use data of normal class to build a data description that captures all characteristics of the data. Data of the other abnormal class are used to refine the obtained data description so that it can better describe the actual data. The one-class classification is broadly applied to many real application domains namely network intrusion, currency validation, user verification in computer systems, medical diagnosis [1] , and machine fault detection [2]. In most of real-world applications of one-class classification, the number of normal data samples is much larger than that of abnormal data samples. The reason is that collecting normal data is inexpensive and easy to measure in comparison to collecting abnormal data. For example in machine fault detection application, normal data can be collected directly under the normal condition of machine while collecting abnormal data requires broken machines. Since the prevalence of one class, in one-class classification, the boundary decision primarily comes from the dominant class while in binary classification the data of both classes are used to construct the boundary decision.

There are various approaches to solving the one-class classification, for example density estimation approach [3] [4] [5], neural network based approach [6] [7] [8],

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 529–536, 2011.
© Springer-Verlag Berlin Heidelberg 2011

and kernel based approach [9] [10] [11] [12]. In this paper, we focus on the kernel based approach that has been proven to be successful. Theoretical analyses of this approach have been proposed in [13] [14]. Following this approach, in geometrical view point, a geometrical shape with the minimal volume is determined in feature space to enclose the normal data samples or to separate those samples from the abnormal data samples. The objective functions of these methods always take into account both general error (GE) and empirical error (EE). The minimization of GE is enforced by minimal volume of geometrical shape. Through minimising GE, we want to certify that the obtained classifier still provides good performance on a separate testing set. One-Class Support Vector Machine (OCSVM) [10] aims at conducting an optimal hyperplane to separate the normal samples from the origin such that the margin is maximised. In Support Vector Data Description (SVDD) [9], an optimal hypersphere is built to include all normal data samples and simultaneously exclude all abnormal data samples. As an extension of SVDD, to increase the chance of accepting abnormal samples in Small Sphere Large Margin (SSLM) [11], the authors proposed the smallest hypersphere with the largest margin. However, the side effect of SSLM is that the spherically boundary decision can accidentally break into the region of normal samples. To overcome this drawback, in Small Sphere Two Large Margin (SS2LM) [12] an optimal sphere with two large margins are constructed.

In all of the above-mentioned kernel techniques, the common strategy for searching the best parameter set is to regularly traverse all possible values of all parameters or to perform a grid parameter selection process. At the end of this process, the normal data samples are assigned to the same trade-off parameter regardless of their contributions. The same thing also happens to the abnormal data samples. However, this assumption is not always true because data samples have different contributions to the construction of hypersphere or hyperplane decision boundary. In this paper, we propose a refinement process performed right after the grid parameter selection process. This refinement process measures the contribution and importance of all data samples by introducing a weight to each data sample. Depending on their contribution and importance, the weights are iteratively updated to get a better model. This refinement process can be applied to SVDD, SSLM, SS2LM to form weighted SVDD, weighted SSLM, and weighted SS2LM. Experimental results show that the proposed weighted models outperform the standard ones.

For the purpose of formulation, we formulate the imbalanced training set as $\{x_1, x_2, ..., x_s\}$ where the first m_1 samples are labeled $+1$ and the remaining $m_2 = s - m_1$ samples are labeled -1. Let us also denote the label of sample x_i as y_i $(i = 1, \ldots, s)$, where $y_i = 1$ $(i = 1, \ldots, m_1)$ and $y_i = -1$ $(i = m_1 + 1, \ldots, s)$.

2 Support Vector Data Description (SVDD)

SVDD [9] aims at determining an optimal hypersphere to include the normal data samples while the abnormal data samples are outside this hypersphere. The optimisation problem is as follows

$$\min_{R,c,\xi} \left(R^2 + C_1 \sum_{i=1}^{m_1} \xi_i + C_2 \sum_{i=m_1+1}^{s} \xi_i \right) \tag{1}$$

subject to

$$\begin{array}{ll} ||\phi(x_i) - c||^2 \leq R^2 + \xi_i & i = 1, \ldots, m_1 \\ ||\phi(x_i) - c||^2 \geq R^2 - \xi_i & i = m_1 + 1, \ldots, s \\ \xi_i \geq 0, \quad i = 1, \ldots, s & \end{array} \tag{2}$$

where R is radius of the hypersphere, C_1 and C_2 are constants, $\xi = [\xi_i]_{i=1,\ldots,s}$ is vector of slack variables, $\phi(.)$ is a transformation from input space to feature space, and c is center of the hypersphere.

For classifying an unknown data point x, the following decision function is used: $f(x) = sign(R^2 - ||\phi(x) - c||^2)$. The unknown data point x is normal if $f(x) = +1$ or abnormal if $f(x) = -1$.

3 Weighted Support Vector Data Description (WSVDD)

The proposed approach can be applied to all of the above-mentioned kernel techniques. In this section, we present the proposed approach applied to SVDD and it can be extended to other techniques.

3.1 WSVDD Formulation

We extend SVDD to conduct our new model. For SVDD, ξ_i stands for the error at sample x_i. An individual weight will be assigned to each data sample and it can be considered as scaling factor of the error $\xi_i \, (i = 1 \ldots s)$. It leads to the following model

$$\min_{R,\xi,c} \left(R^2 + C_1 \sum_{i=1}^{m_1} \lambda_i \xi_i + C_2 \sum_{i=m_1+1}^{s} \lambda_i \xi_i \right) \tag{3}$$

subject to

$$\begin{array}{l} y_i \|\phi\,(x_i) - c\|^2 \leq y_i R^2 + \xi_i, i = 1, \ldots, s \\ \xi_i \geq 0, i = 1, \ldots, s \end{array} \tag{4}$$

Due to minimising the objective function in (3), if the weight λ_i is big then ξ_i should be small. Hence, we can use the weight λ_i to govern the error ξ_i. The Lagrange function is as follows

$$\begin{array}{l} L(R, c, \xi, \alpha, \beta) = R^2 + C_1 \sum\limits_{i=1}^{m_1} \lambda_i \xi_i + C_2 \sum\limits_{i=m_1+1}^{s} \lambda_i \xi_i + \\ \sum\limits_{i=1}^{s} \alpha_i \left(y_i \|\phi(x_i) - c\|^2 - y_i R^2 - \xi_i \right) - \sum\limits_{i=1}^{s} \beta_i \xi_i \end{array} \tag{5}$$

Setting derivatives to zero, we obtain

$$\frac{\partial L}{\partial R} = 0 \Rightarrow \sum_{i=1}^{s} \alpha_i y_i = 1 \tag{6}$$

$$\frac{\partial L}{\partial c} = 0 \Rightarrow c = \sum_{i=1}^{s} \alpha_i y_i \phi(x_i) \tag{7}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \alpha_i + \beta_i = \lambda_i C_1, i = 1, \ldots, m_1 \tag{8}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \alpha_i + \beta_i = \lambda_i C_2, i = m_1 + 1, \ldots, s \tag{9}$$

$$\alpha_i \geq 0, y_i \|\phi(x_i) - c\|^2 \leq y_i R^2 + \xi_i, \alpha_i \left(y_i \|\phi(x_i) - c\|^2 - y_i R^2 - \xi_i \right) = 0 \tag{10}$$

$$\beta_i \geq 0, \xi_i \geq 0, \beta_i \xi_i = 0, i = 1, \ldots, s \tag{11}$$

Substituting (6), (7), (8) and (9) to the Lagrange function, we have

$$\min_{\alpha} \Big(\sum_{i=1}^{s} \sum_{j=1}^{s} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{s} \alpha_i y_i K(x_i, x_i) \Big) \tag{12}$$

subject to

$$\begin{aligned} &\sum_{i=1}^{s} \alpha_i y_i = 1; 0 \leq \alpha_i \leq \lambda_i C_1, i = 1, \ldots, m_1 \\ &0 \leq \alpha_i \leq \lambda_i C_2, i = m_1 + 1, \ldots, s \end{aligned} \tag{13}$$

To compute radius R, we use KKT conditions in (10) and (11) and denote

$$\begin{aligned} SV_p &= \{i : 1 \leq i \leq m_1 \text{ and } 0 < \alpha_i < \lambda_i C_1\} \\ SV_n &= \{i : m_1 < i \leq s \text{ and } 0 < \alpha_i < \lambda_i C_2\} \end{aligned} \tag{14}$$

It is easy to see that

$$R^2 = \frac{1}{n_1} P_1 = \frac{1}{n_2} P_2 \tag{15}$$

where $n_1 = |SV_p|$ and $n_2 = |SV_n|$, P_1 and P_2 can be computed as

$$P_1 = \sum_{i \in SV_p} \|\phi(x_i) - c\|^2 = \sum_{i \in SV_p} \left(K(x_i, x_i) + \|c\|^2 - 2 \sum_{k=1}^{s} y_k \alpha_k K(x_k, x_i) \right) \tag{16}$$

$$P_2 = \sum_{i \in SV_n} \|\phi(x_i) - c\|^2 = \sum_{i \in SV_n} \left(K(x_i, x_i) + \|c\|^2 - 2 \sum_{k=1}^{s} y_k \alpha_k K(x_k, x_i) \right) \tag{17}$$

$$\|c\|^2 = \sum_{i=1}^{s} \sum_{j=1}^{s} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \tag{18}$$

For classification of a new sample x, we calculate the distance between $\phi(x)$ and center c of the hypersphere and then classify x as normal if this distance is less than radius R and as abnormal otherwise. The decision function is of the form

$$\begin{aligned} f(x) &= sign\left(R^2 - \|\phi(x) - c\|^2\right) \\ &= sign\left(R^2 - \|c\|^2 - K(x, x) + 2 \sum_{i=1}^{s} \alpha_i y_i K(x_i, x)\right) \end{aligned} \tag{19}$$

3.2 Refinement Process

This process is performed after the grid parameter selection process of SVDD. Similar to Boosting algorithm [16], we concentrate on the samples that cause error. Since the prevalence of the normal samples, we do not pay attention to the abnormal samples but regularly update the weights of misclassified normal samples. The refinement process iteratively detects the normal samples suffered the error and updates the weights of these samples. We divide the error of misclassified normal samples into two kinds. The first kind of error includes samples locating in the confused region, i.e. the region that contains both the normal and abnormal samples. The second one includes samples residing in the unconfused region (the samples probably locate far away from the abnormal samples). We heuristically found that the decrease of the first kind of error is to more efficiently boost the performance of SVDD than that of the second one. The rational of this heuristic will be explained in the next two sections. By conforming to the above heuristic, we attempt to find out the normal samples overtaking the first kind of error then make weights of these samples double and in the meanwhile reduce half weights of the normal samples in the second kind of error. The algorithm for this process is presented as follows

```
Perform clustering data in the input space:
   Discover clusters that contain both normal and abnormal data
   Denote those clusters as MIXEDCLUS
Perform the grid parameter selection for SVDD
For each sample p, set weight w[p]=1
   For i from 1 to 100 do
     Train data with current trade-off parameters set
     Find out the normal samples p suffered error, denote as POSERROR
     Update weights of all samples in POSERROR
        If p in MIXEDCLUS then w[p]=w[p]*2 Else w[p]=w[p]/2
```

4 Rational of the Proposed Refinement Process

4.1 Proposition of Empirical Error

Theorem 1. *Let us denote solutions of SVDD or WSVDD as* (R, c, ξ) *then* $\xi_i = \max\left\{y_i\left(\|\phi(x_i) - c\|^2 - R^2\right), 0\right\}, i = 1, \ldots, s$. *In case that error occurs at* x_i *then* $\xi_i = y_i\left(\|\phi(x_i) - c\|^2 - R^2\right)$

Proof: For SVDD or WSVDD, referring to the constrains, we have $\xi_i \geq \|\phi(x_i) - c\|^2 - R^2, i = 1, \ldots, m_1$ and $\xi_i \geq R^2 - \|\phi(x_i) - c\|^2, i = m_1 + 1, \ldots, s$

Using the fact that $\xi_i \geq 0$ and we need to minimise $\sum_{i=1}^{n} \xi_i$, we obtain $\xi_i = \max\left\{y_i\left(\|\phi(x_i) - c\|^2 - R^2\right), 0\right\}, i = 1, \ldots, s$. If error happens at x_i then $y_i\left(\|\phi(x_i) - c\|^2 - R^2\right) > 0$. Hence $\xi_i = y_i\left(\|\phi(x_i) - c\|^2 - R^2\right)$. Clearly, we can see that the empirical error $\sum_{i=1}^{n} \xi_i$ is the sum of distances from the misclassified samples to the sphere surface.

We present the advantage of the training WSVDD using a mathematical explanation below.

In (3), denote $\xi'_i = \lambda_i \xi_i (i = 1, \ldots, s)$, we can rewrite (3) and (7) as follows

$$\min_{R,c,\xi}\Big(R^2 + C_1 \sum_{i=1}^{m_1} \xi'_i + C_2 \sum_{i=m_1+1}^{s} \xi'_i \Big) \tag{20}$$

subject to

$$\begin{aligned} &\|\phi(x_i) - c\|^2 \leq R^2 + \xi'_i/\lambda_i, i = 1, \ldots, m_1 \\ &\|\phi(x_i) - c\|^2 \geq R^2 - \xi'_i/\lambda_i, i = m_1 + 1, \ldots, s \\ &\xi'_i \geq 0, i = 1, \ldots, s \end{aligned} \tag{21}$$

It is easy to see that in case x_i is not in any mixed clusters, λ_i is small and it causes that the constraint related to x_i is looser than others. Hence, the constraints related to samples of mixed clusters are tighter than others. It follows that the new model is more concerned to the samples of mixed clusters than the other regions.

5 Experimental Results

We performed our experiment on 22 well-known data sets related to machine fault detection and bioinformatics. Some of them are multi-class data sets. For the purpose of our experiments, we constructed the training sets such that they contained plenty of normal samples and a few of abnormal samples. For each data set, to conduct a training set, we appointed a class as normal class and

placed all data points of this class into the training set. The data of remaining classes would be considered as abnormal data. We randomly picked abnormal data points from the remaining classes and put into the training set so that the ratio between normal and abnormal data in the training set is 9 : 1. We run cross validation with 5 folds for each training set. To ensure the stable performance of the classifiers, we run the classifiers 10 times on each data set and compute mean of accuracies. To get the best model, we use cross validation method with five folds. To compute accuracy, we followed the formula in [15] which is $acc = \sqrt{acc^+ acc^-}$ where acc, acc^+, and acc^- are accuracies over whole training set, positive (normal) class, and negative (abnormal) class, respectively.

As shown in the literature, we chose RBF Kernel function $K(x, x') = e^{-\gamma \|x-x'\|^2}$ where parameter γ is varied over grid $\left\{2^k : k = 2l + 1, l = -8..2\right\}$. For SVDD, trade-off parameter C_1 will be ranged over grid $\left\{2^k : k = 2l + 1, l = -8..2\right\}$ whereas trade-off parameter C_2 will be ranged such that the ratio $\frac{C_2}{C_1}$ will be contained in grid $\left\{\frac{1}{4}.\frac{m_1}{m_2}; \frac{1}{2}.\frac{m_1}{m_2}; \frac{m_1}{m_2}; 2.\frac{m_1}{m_2}; 4.\frac{m_1}{m_2}\right\}$. For OCSVM, parameter ν will be varied in $\{0.1k, 0.01k\}$ where k is an integer number ranging from 1 to 9. For SSLM and SS2LM, parameter ν will be varied in grid $\{10; 30; 50; 70; 90; 110\}$ and parameters ν_1, ν_2 will be taken in $\{0.01; 0.1\}$. Finally, for SS2LM, parameter δ will be slid in grid $\{0.k\nu : k = 0, \ldots, 10\}$. For computation of the weights, we would apply the

Table 1. Experimental results for the 22 data sets

Data set	OCSVM	SVDD	WSVDD	SSLM	WSSLM	SS2LM	WSS2LM
Arrhythmia	67%	64.1%	85.58%	65.28%	82.44%	67.3%	86.8%
Astroparticle	95.8%	94.75%	95.1%	93.15%	94.3%	93.2%	94.7%
Australian	93.15%	94.8%	95.2%	95.3%	96.2%	95.5%	96.63%
Breast Cancer	98.2%	99.18%	99.18%	99.2%	99.5%	99.1%	99.61%
Bioinformatics	68.75%	65.14%	69.10%	67.26%	70.22%	68.3%	71.45%
Biomed	94.37%	96.73%	97.9%	97.15%	98%	97.1%	98.24%
Delf Pump	92.8%	91.58%	91.64%	92.7%	93.46%	93%	93.5%
Diabetes	71.75%	73.25%	73.25%	74.15%	74.2%	74.5%	74.83%
Dna	92.78%	93.27%	95.28%	92.38%	94.72%	91.8%	93.56%
Fourclass	97.26%	99.81%	99.81%	99.85%	99.86%	99.88%	99.9%
Glass	96.24%	94.61%	94.61%	95.38%	95.67%	96.4%	97.31%
Heart	82.36%	86%	86%	86.2%	86.22%	86.53%	86.54%
Hepatitis	87.95%	88.94%	92.65%	89.1%	94.2%	90.34%	95.18%
Ionosphere	99.14%	97.2%	97.72%	97.34%	98.12%	98.1%	98.24%
Letter	96.9%	99.8%	99.9%	99.2%	99.22%	99.12%	99.31%
Sonar	94.58%	94.64%	94.64%	95.23%	95.36%	96.21%	96.23%
Spectf	73.69%	73.23%	73.64%	73.5%	73.7%	73.61%	73.88%
Splice	80.58%	81.73%	81.87%	82.31%	82.41%	83.27%	83.47%
SvmGuide1	95.83%	95.69%	95.99%	96.21%	96.32%	96.4%	96.53%
SvmGuide3	76.4%	78.91%	78.91%	76.25%	76.26%	75.88%	75.92%
Thyroid	98.21%	97.64%	97.69%	98%	98.33%	98.17%	98.21%
Vehicle	81.9%	82.63%	82.92%	84.14%	84.22%	85.31%	85.37%

fuzzy c-means (FCM) clustering algorithm and slide the number of clusters from 1 to 10. Table 1 shows that the new parameter selection strategy outperforms the normal grid parameter selection strategy on the 22 data sets.

6 Conclusion

We have propsed an iterative learning process to refine the trade-off parameters of SVMs. This process needs to be carried out right after the grid parameter selection process of SVMs. The experimental results show that the new approach improves the performance. The new approach can also be applied to other One Class Support Vector Machines such as SVDD, SSLM and SS2LM.

References

1. Campbell, C., Bennett, K.P.: A linear programming approach to novelty detection. Advances in Neural Information Processing Systems 13, 395–401 (2001)
2. Towel, G.G.: Local expert autoassociator for anomaly detection. In: Proc. 17th Int. Conf. on Machine Learning, pp. 1023–1030 (2000)
3. Bishop, C.M.: Novelty detection and neural network validation. In: IEE Proc. of Vision, Image and Signal Processing, pp. 217–222 (1994)
4. Barnett, V., Lewis, T.: Outliers in statistical data, 3rd edn. Wiley (1978)
5. Roberts, S., Tarassenko, L.: A Probabilistic Resource Allocation Network for Novelty Detection. Neural Computation 6, 270–284 (1994)
6. Ritter, G., Gallegos, M.T.: Outliers in statistical pattern recognition and an application to automatic chromosome classification. Pattern Recognition Letters 18(6), 525–539 (1997)
7. Richard, M.D., Lippmann, R.P.: Neural network classifiers estimate Bayesian a posteriori probabilities. Neural Computation 3(4), 461–483 (1991)
8. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press (1996)
9. Tax, D.M.J., Duin, R.P.W.: Support vector data description. Machine Learning 54, 45–56 (2004)
10. Scholkopf, B., Smola, A.J.: Learning with kernels. The MIT Press (2001)
11. Wu, M., Ye, J.: A Small Sphere and Large Margin Approach for Novelty Detection Using Training Data with Outliers. IEEE Trans. Pattern Analysis & Machine Intelligence 31, 2088–2092 (2009)
12. Le, T., Tran, D., Ma, W., Sharma, D.: An Optimal Sphere and Two Large Margins Approach for Novelty Detection. In: Proc. IEEE WCCI, pp. 909–914 (2010)
13. Scott, C.D., Nowak, R.D.: Learning minimum volume sets. Journal of Machine Learning 7, 665–704 (2006)
14. Vert, J., Vert, J.P.: Consistency and convergence rates of one class svm and related algorithm. Journal of Machine Learning Research 7, 817–854 (2006)
15. Lin, Y., Lee, Y., Wahba, G.: Support vector machine for classification in nonstandard situations. Machine Learning 15, 1115–1148 (2002)
16. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: European Conf. on Computational Learning Theory, pp. 23–37 (1995)

Multi-Sphere Support Vector Clustering

Trung Le, Dat Tran, Phuoc Nguyen, Wanli Ma, and Dharmendra Sharma

Faculty of Information Sciences and Engineering
University of Canberra, ACT 2601, Australia
{trung.le,dat.tran,phuoc.nguyen,
wanli.ma,dharmendra.sharma}@canberra.edu.au

Abstract. Current support vector clustering method determines the smallest sphere that encloses the image of a dataset in feature space. This sphere when mapped back to data space will form a set of contours that can be interpreted as cluster boundaries for the dataset. However this method does not guarantee that the single sphere and the resulting cluster boundaries can best describe the dataset if there are some distinctive data distributions in this dataset. We propose multi-sphere support vector clustering to address this issue. Data points in data space are mapped to a high dimensional feature space and a set of smallest spheres that encloses the image of the dataset is determined. This set of spheres when mapped back to data space will form a set of contours that can be interpreted as cluster boundaries. Experiments on different datasets are performed to demonstrate that the proposed approach provides a better cluster analysis than the current support vector clustering method.

Keywords: Cluster analysis, support vector data description, support vector machine, kernel method.

1 Introduction

Clustering in a set of unlabeled data points is to assign to data points labels that identify subgroups in that set [1]. An unsupervised learning algorithm is used to determine those supgroups known as clusters according to a given clustering criterion. K-means [2], fuzzy C-means [1] and fuzzy entropy [3] are some parametric clustering algorithms. Support vector clustering (SVC) [4] is a non-parametric clustering algorithm based on the support vector machine approach [5]. Data points are mapped by means of a Gaussian kernel to a high dimensional feature space and a smallest sphere that encloses the image of those data points is determined. This sphere when mapped back to data space will form a set of contours that can be interpreted as cluster boundaries for that data set. SVC provides an efficient way to deal with outliers and explicit calculations in feature space are not necessary [6]. However this method does not guarantee that the single sphere and the resulting cluster boundaries can best describe the dataset if there are some distinctive data distributions in this dataset. We propose multi-sphere support vector clustering (MSSVC) to address this issue.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 537–544, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Data points in data space are mapped to a high dimensional feature space and a set of smallest spheres that encloses the image of the data is determined using an iterative learning algorithm that ensures that the clustering error is reduced in each iteration. This set of spheres when mapped back to data space will form a set of contours that can be interpreted as cluster boundaries. Experiments on different datasets are performed to demonstrate that the proposed MSSVC method provides a better cluster analysis than the SVC method.

The paper is organised as follows. Section 2 presents Support Vector Data Description (SVDD) which is the framework for both the SVC and MSSVC algorithms. Section 3 presents the SVC algorithm. We propose the MSSVC algorithm in Section 4 and then present experimental results to evaluate MSSVC and compare with SVC in Section 5. Finally a conclusion is given in Section 6.

2 Support Vector Data Description (SVDD)

Let $X = \{x_1, x_2, \ldots, x_n\}$ be the data set. SVDD [7] aims at determining an optimal sphere including all normal data points in this data set X while abnormal data points are not included. The optimisation problem is as follows

$$\min_{R,c,\xi} \left(R^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i \right) \tag{1}$$

subject to

$$||\phi(x_i) - c||^2 \leq R^2 + \xi_i, \qquad \xi_i \geq 0, \quad i = 1, \ldots, n \tag{2}$$

where R is radius of the hypersphere, c is centre of the sphere, $\xi = [\xi_i]_{i=1,\ldots,n}$ is vector of slack variables, ν is a positive constant, $\phi(.)$ is the nonlinear function related to the symmetric, positive definite kernel function $K(x_1, x_2) = \phi(x_1)^T \phi(x_2)$. The centre c and radius R are calculated as follows:

$$\begin{aligned} c &= \sum_{i=1}^{n} \alpha_i \phi(x_i), \qquad \|c\|^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) \\ R^2 &= \frac{1}{|BSV|} \sum_{i \in BSV} \|\phi(x_i) - c\|^2 \\ &= \frac{1}{|BSV|} \sum_{i \in BSV} \left(K(x_i, x_i) + \|c\|^2 - 2 \sum_{k=1}^{n} \alpha_k K(x_k, x_i) \right) \end{aligned} \tag{3}$$

where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]$ is solution of the following optimisation problem:

$$\min_{\alpha} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{n} \alpha_i K(x_i, x_i) \tag{4}$$

$$\sum_{i=1}^{n} \alpha_i = 1, \qquad 0 \leq \alpha_i \leq \frac{1}{\nu n} \quad i = 1, \ldots, n \tag{5}$$

where $BSV = \left\{ i | \, 1 \leq i \leq n, 0 < \alpha_i < \frac{1}{\nu n} \right\}$ is the set of indices of bounded support vectors.

3 Support Vector Clustering (SVC)

SVC employs SVDD with Gaussian kernel $K(x, x') = e^{-\gamma \left\| x - x' \right\|^2}$. The parameters ν and γ govern the shape and the enclosing contours in data space. The number of bounded support vectors increases with increasing ν and the boundary fits more tightly the data. The increase of parameter γ results in the increase of number of bounded support vectors and as a result more clusters are determined.

According to the KKT theorem [4] in SVDD, all bounded support vectors are on the optimal sphere in feature space. When being mapped back to data space, these bounded support vectors become the boundary points of clusters. The assignment is based on the following observation: if two points belong to two different clusters then all paths connecting the two points must exit from the sphere in feature space. This observation leads to the definition of the adjacency matrix as follows:

$$A_{ik} = \begin{cases} 1 & \text{if, for all } y \text{ on the line segment connecting } x_i \text{ and } x_k, R(y) \leq R \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $R(y)$ is distance from $\phi(y)$ to centre of the optimal sphere in feature space and can be computed as follows:

$$R(y) = K(y, y) + \left\| c \right\|^2 - 2 \sum_{k=1}^{n} \alpha_k K(x_k, y) \tag{7}$$

4 Multi-Sphere Support Vector Clustering (MSSVC)

4.1 Problem Formulation

In MSSVC, data points are also mapped by means of a Gaussian kernel to a high dimensional feature space. However a number of smallest spheres that enclose the image of those data points is determined. These spheres when mapped back to data space will form sets of contours that can be interpreted as cluster boundaries for that data set. Consider a set of m spheres $S(c_j, R_j)$ where c_j and R_j are centre and radius of sphere S_j. This set of m spheres is regarded as a good data description if it can enclose all data points and the sum $\sum_{j=1}^{m} R_j^2$ is minimized to provide a minimal general error. Let matrix $U = [u_{ij}]_{n \times m}$, $i = 1, \ldots, n$, $j = 1, \ldots, m$ where u_{ij} denotes the degree of belonging of $\phi(x_i)$ to sphere S_j, $u_{ij} = 0$ if $\phi(x_i)$ is not in S_j and $u_{ij} = 1$ if $\phi(x_i)$ is in S_j.

4.2 Calculating Radii and Centres

Calculating radii and centres is based on the following optimisation problem:

$$\min_{R,c,\xi} \Big(\sum_{j=1}^{m} R_j^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i \Big) \tag{8}$$

subject to

$$\sum_{j=1}^{m} u_{ij} ||\phi(x_i) - c_j||^2 \leq \sum_{j=1}^{m} u_{ij} R_j^2 + \xi_i, \qquad \xi_i \geq 0, \quad i = 1, \ldots, n \tag{9}$$

where $R = [R_j]_{j=1,\ldots,m}$ is vector of radii, ν is a constant, $\xi = [\xi_i]_{i=1,\ldots,n}$ is vector of slack variables. The Lagrange function L is determined as follows

$$L(R, c, \xi, \alpha, \beta) = \sum_{j=1}^{m} R_j^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \alpha_i \Big(||\phi(x_i) - c_{s(i)}||^2 - R_{s(i)}^2 - \xi_i \Big) - \sum_{i=1}^{n} \beta_i \xi_i \tag{10}$$

where $s(i)$ is index of the sphere to which data point x_i belong and satisfies $u_{is(i)} = 1$ and $u_{ij} = 0 \; \forall j \neq s(i)$. Setting derivatives of $L(R, c, \xi, \alpha, \beta)$, we obtain

$$\frac{\partial L}{\partial R_j} = 0 \quad \Rightarrow \sum_{i \in s^{-1}(j)} \alpha_i = 1 \tag{11}$$

$$\frac{\partial L}{\partial c_j} = 0 \quad \Rightarrow c_j = \sum_{i \in s^{-1}(j)} \alpha_i \phi(x_i) \tag{12}$$

$$\frac{\partial L}{\partial \xi_j} = 0 \quad \Rightarrow \alpha_i + \beta_i = \frac{1}{\nu n}, \quad i = 1, \ldots, n \tag{13}$$

$$\begin{aligned} \alpha_i \geq 0, \quad ||\phi(x_i) - c_{s(i)}||^2 - R_{s(i)}^2 - \xi_i \geq 0, \\ \alpha_i \Big(||\phi(x_i) - c_{s(i)}||^2 - R_{s(i)}^2 - \xi_i \Big) = 0 \end{aligned} \tag{14}$$

$$\beta_i \geq 0, \quad \xi_i \geq 0, \quad \beta_i \xi_i = 0 \tag{15}$$

To get the dual form, we substitute (11)-(15) to (10) and obtain the following:

$$\begin{aligned} L &= \sum_{i=1}^{n} \alpha_i ||\phi(x_i) - c_{s(i)}||^2 \\ &= \sum_{i=1}^{n} \alpha_i K(x_i, x_i) - 2 \sum_{i=1}^{n} \alpha_i \phi(x_i) c_{s(i)} + \sum_{i=1}^{n} \alpha_i ||c_{s(i)}||^2 \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{n} \alpha_i K(x_i, x_j) - \sum_{j=1}^{m} \sum_{i \in s^{-1}(j)} \alpha_i \phi(x_i) c_j + \sum_{j=1}^{m} \sum_{i \in s^{-1}(j)} \alpha_i ||c_j||^2 \\
&= \sum_{i=1}^{n} \alpha_i K(x_i, x_j) - 2 \sum_{j=1}^{m} \Big(c_j \sum_{i \in s^{-1}(j)} \alpha_i \phi(x_i) \Big) + \sum_{j=1}^{m} \Big(||c_j||^2 \sum_{i \in s^{-1}(j)} \alpha_i \Big) \\
&= \sum_{i=1}^{n} \alpha_i K(x_i, x_j) - \sum_{j=1}^{m} ||c_j||^2 \\
&= \sum_{j=1}^{m} \Big(\sum_{i \in s^{-1}(j)} \alpha_i K(x_i, x_i) - ||c_j||^2 \Big) \\
&= \sum_{j=1}^{m} \Big(\sum_{i \in s^{-1}(j)} \alpha_i K(x_i, x_i) - || \sum_{i \in s^{-1}(j)} \alpha_i \phi(x_i)||^2 \Big) \\
&= \sum_{j=1}^{m} \Big(\sum_{i \in s^{-1}(j)} \alpha_i K(x_i, x_i) - \sum_{i,i' \in s^{-1}(j)} \alpha_i \alpha_{i'} K(x_i, x_{i'}) \Big) \qquad (16)
\end{aligned}
$$

The result in (16) shows that the optimisation problem in (8) is equivalent to m individual optimisation problems as follows

$$
\min \Big(\sum_{i \in s^{-1}(j)} \alpha_i K(x_i, x_i) - \sum_{i,i' \in s^{-1}(j)} \alpha_i \alpha_{i'} K(x_i, x_{i'}) \Big) \quad j = 1, \ldots, m \qquad (17)
$$

subject to

$$
\sum_{i \in s^{-1}(j)} \alpha_i = 1 \quad \text{and} \quad 0 \leq \alpha_i \leq \frac{1}{\nu n} \quad j = 1, \ldots, m \qquad (18)
$$

After solving all of these individual optimization problems, we can calculate the updating $R = [R_j]$ and $c = [c_j]$, $j = 1, \ldots, m$ using the equations in SVDD.

4.3 Calculating Matrix U

With the radii and centres calculated, there exist m separate spheres $S(c_j, R_j)$, $j = 1, \ldots, m$ in feature space. It can be seen that the map $\phi(x_i)$ of data point x_i is either in a particular sphere or not in any of those spheres.

If $\phi(x_i)$ is in some spheres in the sphere set $J = \{j : \phi(x_i) \in S(c_j, R_j)\}$, $\phi(x_i)$ will be assigned to the closest sphere $S(c_{j_0}, R_{j_0})$ where $j_0 = \arg\min_{j \in J} ||\phi(x_i) - c_j||^2$, and we have $u_{ij_0} = 1$ and $u_{ij} = 0$ if $j \neq j_0$.

If $\phi(x_i)$ is not in any of those spheres, $\phi(x_i)$ will be assigned to sphere $S(c_{j_0}, R_{j_0})$ where $j_0 = \arg\min_j \Big\{ ||\phi(x_i) - c_j||^2 - R_j^2 \Big\}$, and we have $u_{ij_0} = 1$ and $u_{ij} = 0$ if $j \neq j_0$.

4.4 MSSVC Algorithm

The proposed iterative clustering process for MSSVC will run two alternative steps until a convergence is reached as follows

```
Initialise U to fuzzy memberships from clustering the data set in
data space
Repeat the following
     Calculate R and c using U
     Calculate U using R and c
Until a convergence is reached
```

4.5 Clustering Assignment

Similar to SVC, the clustering assignment is based on the following observation: if two data points in a sphere belong to two different clusters then all paths connecting the two points must exit from that sphere in feature space. This observation leads to the definition of the adjacency matrix as follows

$$A_{ik} = \begin{cases} 1 & \text{if, for all } y \text{ on the line segment connecting } x_i \text{ and } x_k, \\ & R_j(y) \leq R_j \text{ for some } j \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where $R_j(y)$ is distance from $\phi(y)$ to centre c_j of the optimal sphere $S(c_j, R_j)$ in feature space.

5 Experimental Results

We consider the shape of the enclosing contours for the Iris data set in data space versus the parameters ν and γ for MSSVC. We apply the Principal Component Analysis method [8] to find out two principal components and perform MSSVC on this two-dimensional Iris data set. Figure 1 demonstrates that if the scale parameter γ of the Gaussian kernel is increased, more support vectors are found and the shape of the boundary in data space fits more tightly the data set, and the enclosing contour splits, forming an increasing number of clusters.

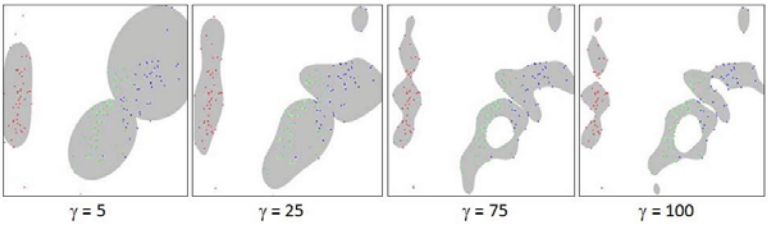


Fig. 1. Clustering Iris data using MSSVC where $\nu = 0.1$ and $\#spheres = 3$

On the other hand, Figure 2 shows that the decrease of parameter ν causes the shape of the boundary more tightly the data and more data points are

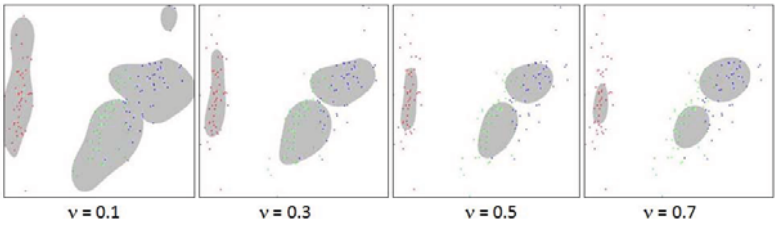


Fig. 2. Clustering Iris data using MSSVC where $\gamma = 25$ and $\#spheres = 3$

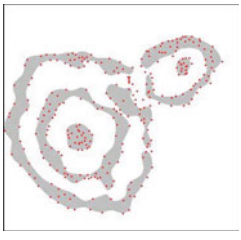


Fig. 3. Clustering using MSSVC, $\gamma = 300, \nu = 0.2$ and $\#spheres = 2$

outside clusters. The parameter ν is regarded as the soft margin that controls the number of outliers.

Similar to SVC, MSSVC is capable of detecting clusters that have complicated distributions. Figure 3 shows the shape of clusters obtained when the number of spheres is set to 2.

5.1 Clustering Examples for SVC and MSSVC

Figure 4 compares clustering results for SVC and MSSVC. There are 3 clusters in that data set and they are identified by MSSVC. However SVC is not capable to identify those clusters although different values of γ and ν were chosen for SVM as seen in Figure 5.

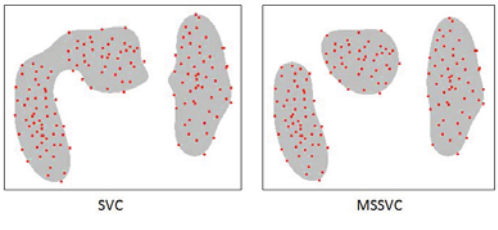


Fig. 4. Clustering using SVC and MSSVC ($\#spheres = 3$) where $\gamma = 100$ and $\nu = 0.1$

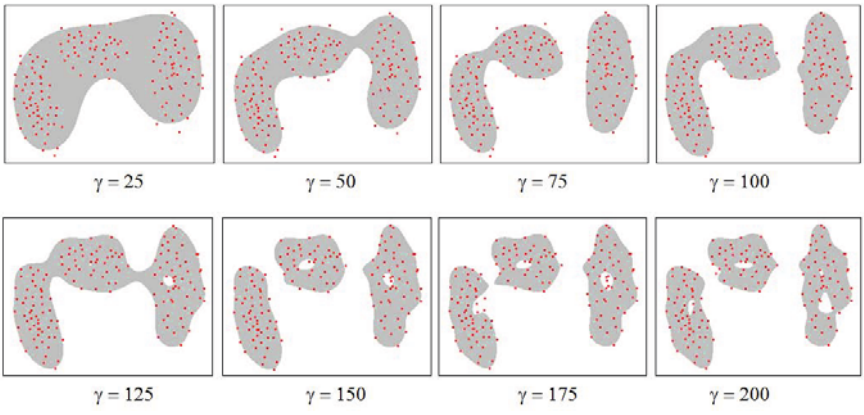


Fig. 5. Clustering using SVC where $\nu = 0.1$

6 Conclusion

We have proposed a new clustering method based on multi-sphere approach to support vector data description. A set of optimal spheres is determined as a good data description for a given data set mapped to a high dimensional feature space. This set of optimal spheres when mapped back to data space form a set of contours that can be interpreted as cluster boundaries. Multi-sphere support vector clustering is capable to discover clusters more powerful than support vector clustering.

References

1. Bezdek, J.C.: A review of probabilistic, fuzzy and neural models for pattern recognition. Journal of Intelligent and Fuzzy Systems 1(1), 1–25 (1993)
2. Duda, R.O., Hart, P.E.: Pattern classification and scene analysis. John Wiley & Sons (1973)
3. Tran, D., Wagner, M.: Fuzzy Entropy Clustering. In: Proceedings of FUZZ-IEEE, vol. 1, pp. 152–157 (2000)
4. Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V.: Support vector clustering. Journal of Machine Learning Research 2, 125–137 (2001)
5. Vapnik, V.: The nature of statistical learning theory. Springer, Heidelberg (1995)
6. Yang, J., Estivill-Castro, V., Chalup, S.K.: Support vector clustering through proximity graph modelling. In: Proceedings of the 9th International Conference on Neural Information Processing, vol. 2, pp. 898–903 (2002)
7. Tax, D.M.J., Duin, R.P.W.: Support vector data description. Machine Learning 54, 45–56 (2004)
8. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann (2005)
9. Le, T., Tran, D., Ma, W., Sharma, D.: Multiple Distribution Data Description Learning Algorithm for Novelty Detection. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS, vol. 6635, pp. 246–257. Springer, Heidelberg (2011)

Testing Predictive Properties of Efficient Coding Models with Synthetic Signals Modulated in Frequency

Fausto Lucena[1,⋆], Mauricio Kugler[2], Allan Kardec Barros[3], and Noboru Ohnishi[1]

[1] Nagoya University, Department of Media Science, Nagoya, Aichi, 464-8603, Japan
lucena@ohnishi.m.is.nagoya-u.ac.jp
[2] Nagoya Institute of Technology, Dept. of Computer Science & Engineering, Nagoya, Japan
[3] Universidade Federal do Maranhão (UFMA), PIB, São Luís, MA, S/N, Brazil

Abstract. Testing the accuracy of theoretical models requires a *priori* knowledge of the structural and functional levels of biological systems organization. This task involves a computational complexity, where a certain level of abstraction is required. Herein we propose a simple framework to test predictive properties of probabilistic models adapted to maximize statistical independence. The proposed framework is motivated by the idea that biological systems are largely biased to the statistics of the signal to which they are exposed. To take these statistical properties into account, we use synthetic signals modulated by a bank of linear filters. To show that is possible to measure the variations between expected (ground truth) and estimate responses, we use a standard independent component algorithm as sparse code network. Our simple, but tractable framework suggests that theoretical models are likely to have predictive dispersions with interquartile (range) error of 4.78% and range varying from 3.26% to 23.89%.

Keywords: Efficient coding theory, information theoretic principles, independent component analysis, synthetic signals, and sparse code.

1 Introduction

Understanding the computational aspects underlying the transformation of sensory and motor stimuli into the nervous system is one of the goals of systems neuroscience [8,6,15]. Special attention has been given to explain how and why organisms process information in the specific way that they do [2,3,1]. A common approach (in computational neuroscience) to analyze these questions has been to use probabilistic models, which can be used to make theoretical predictions about the functional architecture and the structural mechanisms integrating neuronal responses [4,14,9,10]. Researches have shown that a probabilistic neural network optimized to encode natural sounds and images lead to filters whose proprieties resemble the impulse response of the mammalian cochlea and the V1 receptive cells [12,7]. Yet, it has been difficult to devise whether these theoretical predictions reflect the variance of the observed system or the inaccuracies arriving from the model itself. Testing the predictive accuracy of the generative models in terms of standard deviations of their expected results seems to be essential to this issue.

⋆ Corresponding author.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 545–553, 2011.
© Springer-Verlag Berlin Heidelberg 2011

The traditional way to analyze theoretical models has been to use toy examples. However, it is unlikely that all the inherent properties can be properly tested using the toy example approach. As alternative, some studies have suggested to analyze the theoretical response from a data ensemble composed of synthetic signals [12,7]. Still, synthetic data largely differ from the properties of natural stimuli, which the organism has evolved to encode. In the case of synthetic data, the predictive responses are not completely clear [13]. The simplest solution seems to model synthetic datasets, such that their intrinsic statistical properties matches with the biological constraints regulating the organism strategy. This solution takes into consideration that sensory systems are largely biased to the statistics of the signals that they are exposed [11]. This framework involves a computational complexity, where a certain level of abstraction is required. A probable start point, for example, is analyzing the computational strategy subserving the genesis of the intrinsic structures observed in biological networks. One prominent candidate to computational strategy in the nervous system is the principle of efficient coding [3]. It posits that sensory systems are adapted through evolutionary processes to enhance their capacity of transferring information in band-limited conditions. Efficient codes are generally obtained from independent component analysis (ICA) or sparse code algorithms [12,4].

Herein, we focus on testing the predictive accuracy of efficient coding models by analyzing the properties emerging from a neural network adapted to yield a sparse code representation. This task is accomplished by an ICA algorithm that minimizes the mutual dependencies of the neural network output. This analysis is carried out by maximizing the non-Gaussianity of a proposed dataset, whose signals are modulated in frequency to simulate intrinsic properties embedded in the code.

2 Efficient Coding as a Sparse Code Neural Network

A traditional view of sparse coding assumes that a few number of neurons are activated at the same time. Sparseness is thought to be directly proportional to statistical independency, such that increasing independence enhance sparseness [16]. A sparse code model assumes that a random vector $\mathbf{x}$ can be expressed as a linear combination of basis functions $\mathbf{a}_i$ (as intrinsic structures), activated by a set of sparse (code) coefficients s_i:

$$\mathbf{x} = \sum_i \mathbf{a}_i s_i. \tag{1}$$

The goal of a neural network adapted to sparse code learning is to enhance the informational capacity of the signal (being encoded) by reducing the statistical redundancy of the code. This problem is closely related to ICA models, whose algorithms are adapted to minimize the statistical dependence of the sources (independent components). One can, therefore, use a generative model based on ICA to estimate sparse codes. Using standard ICA representation, the sparse code problem can be posed using the following matrix-vector representation:

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{2}$$

where the vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ represents the observed random values and $\mathbf{s} = (s_1, s_2, \ldots, s_n)^{\mathrm{T}}$ the vector containing the sparse code. The relationship between

$\mathbf{x}$ and $\mathbf{s}$ is mapped using the matrix $\mathbf{A}$, whose columns are given by the basis functions $\mathbf{a}_i$. In this view, the transformation goal of the model is to determine a matrix $\mathbf{W}$ so that we can estimate $\hat{\mathbf{s}} = \mathbf{W}\mathbf{x}$ when $\mathbf{W} = \mathbf{A}^{-1}$.

One proposed way to solve this problem is using FastICA. The first step on FastICA is whitening the vector $\mathbf{x}$ using a matrix $\mathbf{V}$ to obtain $\mathbf{z} = \mathbf{V}\mathbf{x} = \mathbf{V}\mathbf{A}\mathbf{s}$, whose general solution is given by $\mathbf{y} = \mathbf{W}\mathbf{z}$. Therefore, it is easy to see that if the matrix $\mathbf{W}$ is equal to $(\mathbf{V}\mathbf{A})^{-1}$, then one can obtain $\mathbf{y} = \mathbf{s}$. This standard ICA algorithm maximizes information using approximations of negentropy as a measure of nongaussianity. Negentropy $J(\mathbf{y})$ can be expressed using the Kullback-Leibler divergence (KL) when we consider the Gaussian probability density as target reference. That is,

$$J(\mathbf{y}) = KL(p(y))||p(y_{\text{gauss}})) \tag{3}$$

$$= \int p(y) \log \frac{p(y)}{p(y_{\text{gauss}})} dy \tag{4}$$

$$= \int p(y) \log p(y) dy - \int p(y) \log p(y_{\text{gauss}}) dy \tag{5}$$

$$= H(\mathbf{y}_{\text{gauss}}) - H(\mathbf{y}) \tag{6}$$

The previous equation represents the Kullback-Leibler divergence between a random vector $\mathbf{y}$ with density $p(y)$ and gaussian random vector $\mathbf{y}_{\text{gauss}}$ with density $p(y_{\text{gauss}})$, whose correlation and variance matrix is identical to $\mathbf{y}$. The intuitive idea behind negentropy is that $\mathbf{y}_{\text{gauss}}$ possesses the largest entropy $[H(.)]$ among random variables with identical variance. Negentropy is also a nonnegative measure that is zero if and only if $p(y_{\text{gauss}}) = p(y)$. The Kullback-Leibler divergence shows that the least the gaussian distribution is, the most structured or "spiky" is the distribution. In brief, it is possible to obtain $\mathbf{W}$ after updating each row $\mathbf{w}_i^{\mathrm{T}}$ by deriving a fixed point-iteration using negentropy as [5]:

$$\mathbf{w}_i \leftarrow E\{\mathbf{z}g(\mathbf{w}_i^{\mathrm{T}}\mathbf{z})\} - E\{\mathbf{z}g'(\mathbf{w}_i^{\mathrm{T}}\mathbf{z})\}\mathbf{w}_i, \tag{7}$$

$$\mathbf{W} \leftarrow (\mathbf{W}\mathbf{W}^{\mathrm{T}})^{-\frac{1}{2}}\mathbf{W}. \tag{8}$$

3 Synthetic Signals Modulated in Frequency

How can we test the predictive accuracy of probabilistic models, such as sparse code networks? Synthetic signals are probably the best solution, because they allow one to test a large amount of possible combinations. But it is unlikely that ensembles composed of sparse pixels, non-orthogonal Gabor functions, gratings [12], as well as random noise [7] can mimic the structure underlying natural and biological stimuli. One must consider that the informational capacity of a neural code largely depends on the behavioral significance of the stimuli nature [13], not only on the sparse structure underlying the input data. To verify this hypothesis, we design a test data compose of sparse structures modulated by a bank of filters. The proposed framework can be understood as follows. Let $\mathbf{p}$ be a random vector composed of N samples drawn from a

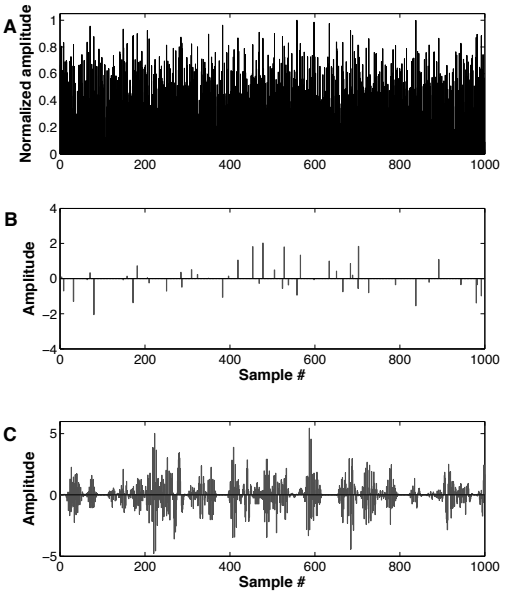


Fig. 1. Illustrative example of synthetic signals obtained from the proposed framework. (A) An interval of 1,000 samples drawn from a normal distribution whose amplitude was normalized between 0 and 1. (B) A sparse sample interval drawn from a normal distribution, whose sample location is obtained from the threshold T using (A). (C) Sample interval modulated in frequency, which is obtained after adding several responses of a bank of filters to the sparse sample interval (B). Although (A-C) are limited to 1,000 samples, we have originally trained our neural network with 100,000 samples.

normally distributed distribution, where $\mathbf{p} = (p_1, \ldots, p_N)$ is normalized to have values ranging from 0 to 1 (Fig. 1A). From this random vector, we select a sparse number of samples p_i by using a threshold (T). Again, let us assume another random vector $\mathbf{q}$ of same distribution and sample size, but drawn from n select sample positions of the previous random vector $\mathbf{p}$ (Fig. 1B). The vector $\mathbf{q} = (q_1, \ldots, q_N)$ can be mathematically expressed as:

$$q_i = \begin{cases} n & \text{if } |p_i / \arg\max_p(\mathbf{p})| \geq T \\ 0 & \text{if } |p_i / \arg\max_p(\mathbf{p})| < T \end{cases}. \tag{9}$$

Using the random vector $\mathbf{q}$, we can generate a synthetic data ensemble by modulating $\mathbf{q}$ with a bank of linear filters $h_1, \ldots, h_K$. The specific response of a filter j is described as,

$$m_j(i) = \sum_{\tau=0}^{N-1} q(\tau) h_j(i - \tau). \tag{10}$$

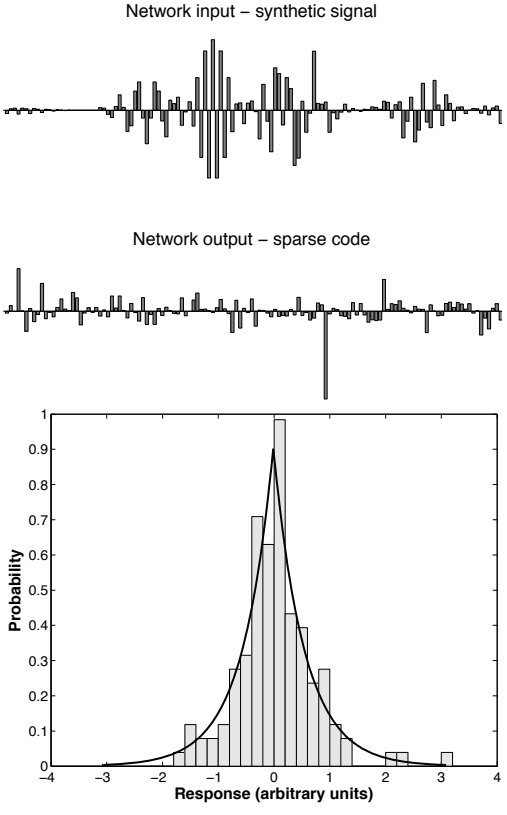


Fig. 2. Sparse coding of synthetic signals modulated in frequency. The bar chart on the top side of the figure illustrates an interval of 128 consecutive samples drawn from synthetic signals. It composes one of the frame windows used to train the sparse coding neural network. The synthetic signals are adapted through an adaptive process of optimization that maximizes the non-Gaussianity of the data ensemble, yielding an output represented by the bar chart (middle side). The two-side distribution of the illustrated network output unit shows a non-Gaussian response that has a sharp peak and heavy tails (bottom side), mostly consistent with a sparse representation. The resulting network output has a sparse representation that in theory depict the degree of active cells involved in coding information. The histogram was fitted using a Laplacian distribution (continuous black line).

4 Estimating Sparse Codes from Synthetic Signals

The first step to test the predictive accuracy of efficient coding models is to generate a data ensemble using the framework described in Section 3. Specifically, we used a vector containing $100,000$ samples with $T = 0.8$ and select a bank of (bandpass) filters with a constant bandwidth of 0.05 Hz shifting in frequency from 0.0 to 0.5 Hz. This procedure is repeated M $(= 18)$ times in total to yield a random vector $\mathbf{f}$ composed of sparse modulated intervals after adding all the filter responses (Fig. 1C), $\mathbf{f} = \sum_{j=1}^{M} m_j$.

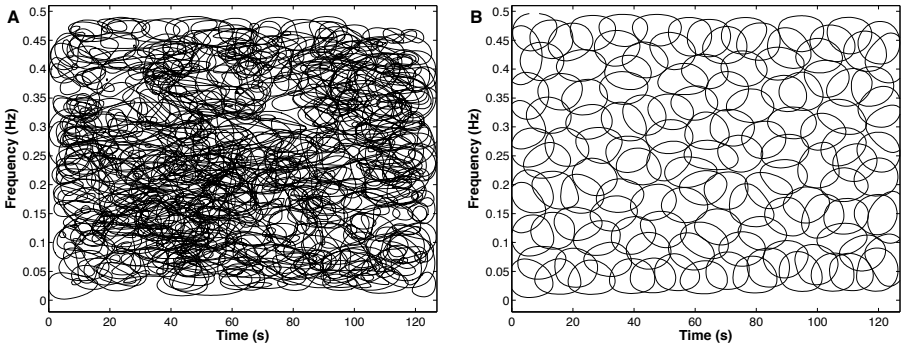


Fig. 3. Joint time and frequency distribution analysis. Contour plot of 128 basis functions corresponding to 95% of the energy from top to bottom. (A) Before learning in whitened space. (B) After learning (optimized with 3,000 iterations).

The second step is to learn the underlying structure of the data ensemble $\mathbf{f}$. The data ensemble $\mathbf{x}$ is obtained by subtracting the mean of the synthetic signal $\mathbf{f}$ and dividing it in non-overlapping intervals (=781) containing 128 samples. Using this data ensemble, a 128×128 covariance matrix is computed and the data whitened, as described in Section 2. We train the adapted ICA neural network according to (7), where $g(.)$ is the hyperbolic tangent $[\tanh(.)]$ and $g'(.)$ is the first derivative of $g(.)$. The initial weight matrix $\mathbf{W}$ is initialized with an identity matrix, which allows a direct search from the maximum variance of the ensemble (as given by a principal component solution). Note, however, that the matrix $\mathbf{W}$ could also be initialized using a random matrix. The matrix $\mathbf{W}$ was updated $3,000$ times to adapt the matrix to yield sparse codes structures, as illustrated in Fig. 2. After learning, the transformation between $\mathbf{A}$ and $\mathbf{W}$ is mapped according to $\mathbf{A} = \mathbf{E}_n\mathbf{D}_n^{1/2}\mathbf{W}^{\mathrm{T}}$, where $\mathbf{D}_n$ is the diagonal matrix represented by the n largest eigenvectors obtained from the correlation matrix ($E\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\} = \mathbf{E}\mathbf{D}\mathbf{E}^{\mathrm{T}}$) and $\mathbf{E}_n$ the corresponding matrix of the eigenvectors (as columns).

5 Results

Figure 2 depicts how the sparse neural network can adapt the input synthetic signal into a sparse code representation. It is easy to see that the sample intervals used as example (top side of Fig. 2) have their structure modified (middle side of Fig. 2), such that only few coefficients are represented by large intensities. This result shows that the algorithm was able to maximize the information of the *synthetic data* from a neural network (based on ICA) optimized to sparse coding learning. To quantitatively confirm whether or not the algorithm yields sparse codes, we have also analyzed the corresponding two-side distribution of the network output. As we can see at bottom side of Fig. 2, the network output has long tails and sharp peak analogous to a sparse representation that follows a Laplacian density (tick line). The important question, however, is if the basis functions retain their design properties after learning. The fist similarity can be observed in Fig. 3.

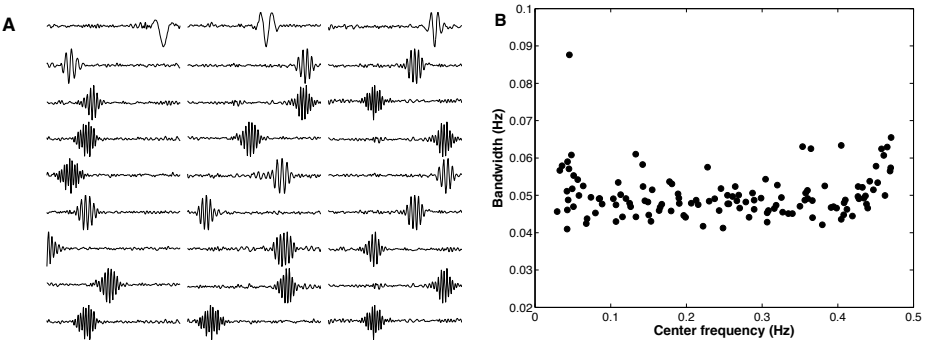


Fig. 4. Bias test from synthetic signals modulated in frequency. (A) A small set of optimized basis organized from lower to higher center frequency (resonant frequency). (B) Predicted bandwidth obtained from 3dB below maximum amplitude spectrum.

Although expected, it is remarkable to observe that the basis functions are spanned, so that they completely cover the time and frequency plane after learning (as shown in Fig. 3B). A result that was not obvious from the analysis of the initial values attributed to the matrix $\mathbf{A}$ (Fig. 3A). This illustrates that the basis functions are covering all the frequencies (0.0–0.5 Hz) found in the linear bank of filters used to modulate the input data. In this case, however, it is not clear if the basis functions have a constant bandwidth of 0.05 Hz similarly to the bank of linear bandpass filters, which subserve the genesis of the synthetic dataset.

To verify this aspect, we analyze the basis functions emerging from the columns of the matrix $\mathbf{A}$ (Fig. 4A). One form is performing an analysis of the properties derived from the basis functions. A straightforward way is using bandwidth, which can be directly computed from the basis functions. The bandwidth (BW) can be obtained by measuring the difference between the lower (f_l) and higher (f_h) cut-off frequency points that are located at -3dB of the maximum resonant peak (BW = f_h - f_l). We have tested the predictive properties of the code by measuring the accuracy of the estimated bandwidth when compared to the expected ground truth (0.05 Hz). As shown in Fig. 4B, the estimated bandwidth is centered around 0.05 Hz. After repeating this procedure with 50 different randomly datasets in which the bandwidth was set to 0.05 Hz, the resulting interquartile (range) yields an error value of 4.39% with a range of 3.26% to 23.89%.

6 Discussion and Conclusion

The observed discrepancies arriving from a (population) code can be either caused by inaccuracies of the model or thought as natural variances that are inherent to the signal ensemble (which is being encoded). By constraining the bandwidth of the bandpass filters to 0.05 Hz, we tested the predictive accuracy of probabilistic models, whose code largely depends on the characteristics underlying the input data. Our results suggest that the response properties arriving from theoretical models have variances that are

uncorrelated with the system under analysis. Therefore, one can suggest that they are caused by inaccuracies arriving from the model itself, not from the data ensemble. This is a curious result. Most of the algorithms used to extract "neural" features from natural stimuli are tested on their capacity of making predictions (about biological systems) and little or no attention has been given to the predictive accuracy of the code. In our case, we have used the FastICA based negentropy, which is very attractive algorithm due to its "computational cost". However, it lacks of a proper (stopping) criteria for the optimization problem, which can cause inaccuracies in the estimated code. It is expected that neural networks that use stochastic gradient algorithms as learning rules tends to be robust to discrepancies that might appear in this models.

Our results are important in the context of (theoretical) neural processing. They drawn some new perspectives on how self-organzing neural networks adapted to learn sparse codes can have discrepancies from their expected results. It would be interesting to test the proposed framework with others sparse code networks (e.g., sparsenet, bayesian models, and informax) to compare the predictive properties emerging from the synthetic data ensemble. Although we have only tested the predictive accuracy by modulating the signal as Fourier-like basis (bandwidth is constant along the center frequency), we believe that the model can be easily extended to design ensemble signals using a bank of filters based on Gabor and wavelet transforms. In more general terms, the proposed framework can be used to evaluate the quality of predictive properties and eventually correct their deviations, which can be one advantage when the code characteristics are unknown.

In conclusion, our results suggests that predictive properties emerging from the population code (learned by the network) presents deviations from their central tendency. Therefore, the response properties arriving from theoretical models should be taken with caution when compared to physiological data.

References

1. Atick, J.J., Redlich, A.N.: Towards a theory of early visual processing. Neural Comput. 2, 308–320 (1999)
2. Attneave, F.: Some informational aspects of visual perception. Psychol. Rev. 61(3), 183–193 (1954)
3. Barlow, H.B.: Possible principles underlying the transformation of sensory messages. In: Rosemblum, M.G. (ed.) Sensory Communication, pp. 217–234. MIT Press, Cambridge, MA (1961)
4. Bell, A., Sejnowski, T.J.: The 'independent components' of natural scenes are edge filters. Vision Research 37, 3327–3338 (1997)
5. Hyvärinen, A., Oja, E.: A fast fixed-point algorithm for independent component analysis. Neural Comput. 9(7), 1483–1492 (1997), http://dx.doi.org/10.1162/neco.1997.9.7.1483
6. Laurent, G.: A systems perspective on early olfactory coding. Science 286(5440), 723–728 (1999)
7. Lewicki, M.S.: Efficient coding of natural sounds. Nat. Neurosci. 5(4), 356–363 (2002)
8. Linsker, R.: Perceptual neural organization: some approches based on network models and information-theory. Annual Review of Neuroscience 13, 257–281 (1990)

9. Lucena, F., Barros, A.K., Ohnishi, N.: Emergence of autonomic transfer properties by learning efficient codes from heartbeat intervals. In: IEICE Tech. Rep. NC2010-153, vol. 110, pp. 153–158. Tokyo (March 2011)
10. Lucena, F., Barros, A.K., Principe, J.C., Ohnishi, N.: Statistical coding and decoding of heartbeat intervals. PLoS One 6(6), e20227 (2011)
11. Machens, C.K., Gollisch, T., Kolesnikova, O., Herz, A.V.M.: Testing the efficiency of sensory coding with optimal stimulus ensembles. Neuron. 47(3), 447–456 (2005)
12. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381(6583), 607–609 (1996)
13. Rieke, F., Bodnar, D.A., Bialek, W.: Naturalistic stimuli increase the rate and efficiency of information transmission by primary auditory afferents. Proc. Biol. Sci. 262(1365), 259–265 (1995)
14. Schwartz, O., Simoncelli, E.P.: Natural signal statistics and sensory gain control. Nature Neuroscience 4, 819–825 (2001)
15. Simoncelli, E.P., Olshausen, B.A.: Natural image statistics and neural representation. Annu. Rev. Neurosci. 24, 1193–1216 (2001)
16. Vinje, W.E., Gallant, J.L.: Sparse coding and decorrelation in primary visual cortex during natural vision. Science 287(5456), 1273–1276 (2000)

A Novel Neural Network for Solving Singular Nonlinear Convex Optimization Problems*

Lijun Liu, Rendong Ge, and Pengyuan Gao

School of Science, Dalian Nationalities University, Dalian 116600, P.R. China
cmliulj@126.com, bgrbgg@163.com

Abstract. Singular nonlinear convex optimization problems have been received much attention in recent years. Most existing approaches are in the nature of iteration, which is time-consuming and ineffective. Different approaches to deal with such problems are promising. In this paper, a novel neural network model for solving singular nonlinear convex optimization problems is proposed. By using LaSalle's invariance principle, it is shown that the proposed network is convergent which guarantees the effectiveness of the proposed model for solving singular nonlinear optimization problems. Numerical simulation further verified the effectiveness of the proposed neural network model.

Keywords: Neural Networks, Singular Nonlinear Optimization, Convergence.

1 Introduction

Tank and Hopfield in 1986 first proposed a neural network for linear programming that was mapped onto a closed-loop circuit [1]. Although the Tank–Hopfield network has a drawback that its equilibrium point may not be an exact solution of the original problem, their pioneering work has inspired many researchers to develop other neural networks for solving linear and nonlinear optimization problems (see [2,3] and the references therein). Kennedy and Chua extended and improved the Tank–Hopfield network by developing a neural network with a finite penalty parameter for solving nonlinear programming problems [2].

Bouzerdoum and Pattison [3] presented a neural network for solving convex quadratic optimization problems with only bounded constraints. Liang [4] and Xia [5] presented neural networks for solving nonlinear convex optimization with bounded constraints and box constraints, respectively. Xia and Wang [6,7,8,9] successfully developed several neural networks for solving linear and quadratic convex programming problems, monotone linear complementary problems, and a class of monotone variational inequality problems. Recently, projection neural networks for solving monotone variational inequality problems are developed in [11,12,13] and recurrent neural networks for solving non-convex optimization problem have been also studied. For example, two neural network models for unconstrained non-convex optimization were presented in [14], and a neural network model for non-convex quadratic optimization was presented in [15].

* This work is partially supported by National Natural Science Foundation of China under Grant 61002039 and the Fundamental Research Funds for the Central Universities DC10040121.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 554–561, 2011.
© Springer-Verlag Berlin Heidelberg 2011

However, little attention was paid to the problem of singular convex optimization problems in the field of neural network. The traditional numerical methods for solving such singular problem can be found in [16,17]. In this paper we present a novel neural network for solving singular nonlinear convex optimization problems.

This paper is organized as follows. In Section 2, the nonlinearly singular convex optimization problem and its equivalent formulations are described. In Section 3, a recurrent neural network model is proposed to solve such singular nonlinear optimization problems. Global convergence of the proposed neural network is obtained. Finally, In Section 4, simulation results are presented, which further validate the effectiveness of the proposed neural network.

2 Problem Formulation and Neural Design

Assume that $f(x) : R^n \rightarrow R$ is convex functionconsidering the following unconstrained convex programming problem

$$\min_{x \in R^n} f(x) . \tag{1}$$

Let $x^*$is a unique optimal solution to (1). We will discuss the solution of (1) under the following assumptions.

Assumption A1. *$f(x)$ is both strictly convex and four times continuous differentiable. For optimum point x^*, there exists $v \in R^n$ such that $rank(\nabla^2 f(x^*)) = n - 1$ and $Null(\nabla^2 f(x^*)) = \{v\}$.*

Assumption A2. *For $x \neq x^*$, there exists $u^T \nabla^2 f(x) u > 0$ for any nonzero $u \in R^n$. Moreover, $\| \nabla^2 f(x) \|$ and $\| \nabla^3 f(x) \|$ are all uniformly bounded.*

Assumption A3. *For any $v \in Null(\nabla^2 f(x^*))$, the quantity*

$$\nabla^4 f(x^*) v^4 \triangleq v^T \nabla^2 (v \nabla^2 f(x^*) v) v > 0$$

. (The reason for this assumption can be found, for example, in [16].)

Lemma 1. *For any $p \in R^n$ and $p^T v \neq 0$, $v \in Null(\nabla^2 f(x^*))$, $(\nabla^2 f(x) + p^T p)$ is nonsingular at x^*.*

Define function $F(x)$ as follows

$$F(x) = f(x) + \lambda h(x),$$

where $h(x) = \mu(x) \nabla^2 f(x) \mu(x)$ and $\mu(x) = (\nabla^2 f(x) + pp^T)^{-1} q$ for $q \neq 0$ and $p^T v \neq 0$.

As the hessian matrix of $f(x)$ at x^* is singular, it is impossible to obtain convergence result by conventional optimization algorithm, (see [16] and [17]). In order to overcome this difficulty, we will deal with equivalent unconstrained convex optimization problem with respect to $F(x)$ defined above, i.e.,

$$\min_{x \in R^n} F(x) \tag{2}$$

For function $F(x)$, we have the following lemmas.

Lemma 2. *For any λ >0, the hessian matrix $\nabla^2 F(x^*)$ is positive definite. Moreover, if $\lambda > 0$ is small enough, then $\nabla^2 F(x)$ is positive definite for any $x \in R^n$.*

Proof. This conclusion can be proved easily according to the results in [16] under Assumption 2. Thus the proof is omitted here for the sake of saving space.

The following two conclusions are obvious. The proof is omitted here.

Lemma 3. *x^* is a solution of (1) if and only if x^* is a solution of (2).*

Considering the trouble caused by computing the matrix inverse, we turn optimization question (2) into the following equivalent constrained optimization problem

$$\begin{cases} \min g(x,y) = f(x) + \lambda y^T \nabla^2 f(x) y \\ s.t. (\nabla^2 f(x) + pp^T) y = q. \end{cases} \tag{3}$$

Its Karush-Kuhn-Tucker condition are summarized as follows,

$$\begin{cases} \nabla f(x) + \lambda \nabla^3 f(x) yy + \nabla^3 f(x) yz = 0, \\ 2\lambda \nabla^2 f(x) y + (\nabla^2 f(x) + pp^T) z = 0, \\ (\nabla^2 f(x) + pp^T) y = q. \end{cases} \tag{4}$$

By Assumption A3., it is easy to know that the function $g(x,y)$ is convex. Based on the Second Order Sufficient Conditions, the KKT point $(\hat{x}, \hat{y})$ of the equation (4) is a unique optimal solution of the optimization question (2). We now establish neural network solution for constrained problem (3).

First, let's define a augmented Lagrange function of (3) as follows

$$\begin{aligned} L(x,y,z) = f(x) + \lambda y^T \nabla^2 f(x) y + z^T [(\nabla^2 f(x) + pp^T) y \\ - q] + \frac{k}{2} \parallel (\nabla^2 f(x) + pp^T) y - q \parallel^2, \end{aligned} \tag{5}$$

where $k > 0$ is a penalty parameter, and $z \in R^n$ is an approximation of the Lagrange multiplier vector.The following is the results in [18].

Lemma 4. *$(\bar{x}, \bar{y}, \bar{z})$ is a stationary point for (4) if and only if for any $k > 0$, $(\bar{x}, \bar{y}, \bar{z}, \bar{z})$ is a stationary point for augmented Lagrange function (5). Moreover, we have*

$$L(\bar{x}, \bar{y}, \bar{z}) = g(\bar{x}, \bar{y}) \tag{6}$$

By Assumption A1.-A3.,the discussing above and Theorem 5 and Theorem 9 in [18], we have that there is a $k_0 > 0$ such that for any $k > k_0$,if $c^* = (x^*, y^*, z^*)$ is an optimal solution of the augmented lagrange function (5), then (x^*, y^*) is an optimal solution of the problem (3) and

$$\min_{x,y,z \in R^n} L(x,y,z) = g(x^*, y^*) = f(x^*)$$

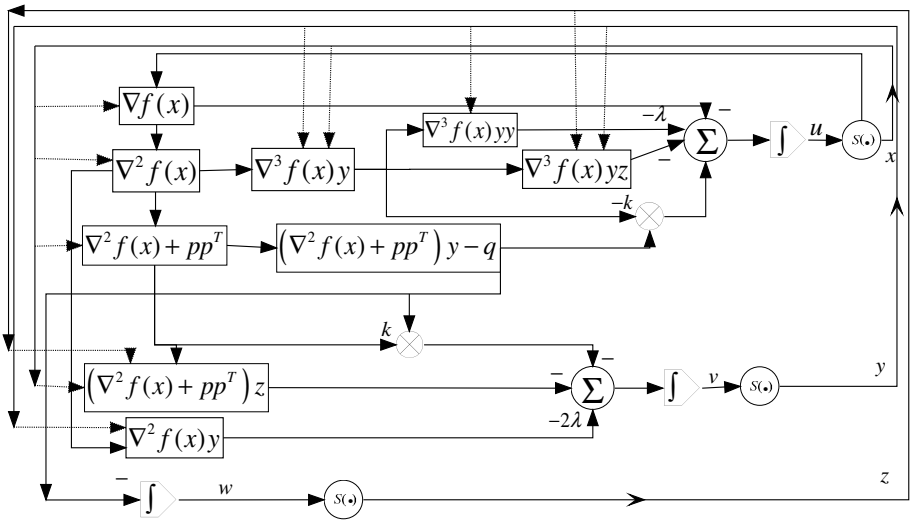


Fig. 1. Logical graph of the proposed neural network model

3 Stability Analysis

By the Lagrange function defined as above, we can describe the neural network model by the following nonlinear dynamic system, for solving (3). The logical graph is shown in Fig.1.

$$\begin{cases} \dfrac{du}{dt} = -\nabla_x L(x,y,z) \\ \quad = -\nabla f(x) - \lambda \nabla^3 f(x)yy - \nabla^3 f(x)yz - k\nabla^3 f(x)y(\nabla^2 f(x) + pp^T)y - q) \\ \dfrac{dv}{dt} = -\nabla_y L(x,y,z) \\ \quad = -2\lambda\nabla^2 f(x)y - (\nabla^2 f(x) + pp^T)z - k(\nabla^2 f(x) + pp^T)((\nabla^2 f(x) + pp^T)y - q) \\ \dfrac{dw}{dt} = -\nabla_z L(x,y,z) = -(\nabla^2 f(x) + pp^T)y + q \\ x_i = s(u_i), \quad i = 1,2,\ldots,n \\ y_j = s(v_j), \quad j = 1,2,\ldots,n \\ z_k = s(w_k), \quad k = 1,2,\ldots,n \end{cases} \tag{7}$$

where

$$\nabla f(x) = (f_1(x), f_2(x), \cdots, f_n(x))^T,$$
$$\nabla^3 f(x)y = (\nabla^2 f_1(x)y, \nabla^2 f_2(x)y, \cdots, \nabla^2 f_n(x)y)^T,$$

and the activation function $s(\cdot)$ is continuously differentiable and satisfy that $s'(\cdot) > 0$.

It is easy to see that the optimal solution (x^*, y^*, z^*) of (5) is an equilibrium point of network (7). Inverse,if (x^*, y^*, z^*) is a equilibrium point of network (7), it must be equilibrium point of original problem (5).

Now we are ready to establish stability and convergence results of network (7).

Theorem 1. *Assume that $f(x) : R^n \to R$ is strictly convex and the fourth differentiable. If the initial point (x_0, y_0, x_0) is chosen in neighborhood about the equilibrium point, then the proposed neural network of (7) is stable in the sense of Lyapunov and globally convergent to the stationary point (x^*, y^*, z^*), where x^* is the optimal solution of (1).*

Proof. Define function $V : \Omega \to R$ as follows,

$$V(x(t), y(t), z(t)) = L(x(t), y(t), z(t)) - f(x^*).$$

We will show that $V(u)$ is a suitable Lyapunov function for dynamic system (7). It is evident that

$$V(x(t), y(t), z(t)) > 0$$

for $(x(t), y(t), z(t)) \neq (x^*, y^*, z^*)$ and $V(x^*, y^*, z^*) = 0$.

Furthermore, there exists that

$$\begin{aligned}
\frac{dV}{dt} &= \sum_{i=1}^{n} \left(\frac{\partial V}{\partial x_i} \cdot \frac{dx_i}{dt} + \frac{\partial V}{\partial y_i} \cdot \frac{dy_i}{dt} + \frac{\partial V}{\partial z_i} \cdot \frac{dz_i}{dt}\right) \\
&= \sum_{i=1}^{n} \left(\frac{\partial V}{\partial x_i} \cdot \frac{dx_i}{du_i} \cdot \frac{du_i}{dt} + \frac{\partial V}{\partial y_i} \cdot \frac{dx_i}{dv_i} \cdot \frac{dv_i}{dt} + \frac{\partial V}{\partial z_i} \cdot \frac{dx_i}{dw_i} \cdot \frac{dw_i}{dt}\right) \\
&= \sum_{i=1}^{n} \left(\frac{\partial V}{\partial x_i} \cdot s'(u_i) \cdot \frac{du_i}{dt} + \frac{\partial V}{\partial y_i} \cdot s'(v_i) \cdot \frac{dv_i}{dt} + \frac{\partial V}{\partial z_i} \cdot s'(w_i) \cdot \frac{dw_i}{dt}\right) \\
&= [\nabla_x L(x, y, z)]^T G'_u \frac{du}{dt} + [\nabla_y L(x, y, z)]^T G'_v \frac{dv}{dt} + [\nabla_z L(x, y, z)]^T G'_w \frac{dw}{dt} \\
&= -[\nabla_x L(x, y, z)]^T G'_u \nabla_x L(x, y, z) - [\nabla_y L(x, y, z)]^T G'_v \nabla_y L(x, y, z) \\
&\quad - [\nabla_z L(x, y, z)]^T G'_w \nabla_z L(x, y, z) \\
&\leq 0
\end{aligned} \tag{8}$$

At the same time, we obtained from (8) that

$$\frac{dV}{dt}(x^*, y^*, z^*) = 0$$

where $G_u = \mathrm{diag}(s(u_1), s(u_2), \cdots, s(u_n))$.

Consequently$V(x(t), y(t), z(t)) = L(x(t), y(t), z(t)) - f(x^*)$ is Lyapunov function, and by (7) and (8), it is evident that

$$\frac{dV}{dt} = 0 \Leftrightarrow \frac{du}{dt} = 0, \frac{dv}{dt} = 0, \frac{dw}{dt} = 0.$$

So the neural network model (7) is asymptotically stable according to the Lyanpunov theory. Therefore, when the initial point (x_0, y_0, x_0) is chosen about the equilibrium point, the set $\{(x(t), y(t), z(t))|t \geq t_0\}$ is bounded. Using LaSalle's invariance principle, the trajectory of the neural network (7) $\{(x(t), y(t), z(t))\}$ will converge to the maximum invariant subset of the following set

$$E = \left\{ (x, y, z) \in S | \ \frac{dV}{dt} = 0 \right\}.$$

Assume again that $X^* = \{x|(x, y, z) \in E\}$, then we have

$$\lim_{t \to \infty} \text{dist}(x(t), X^*) = 0.$$

In special, when $X^* = \{x^*\}$,we have

$$\lim_{t \to \infty} x(t) = x^*$$

The proof is completed.

4 Numerical Example

Consider unconstrained optimization problem as follows,

$$\min_{x \in R^2} f(x) = x_1^4 + x_1^2 + x_2^4 \ . \tag{9}$$

The Hessian of this objective function at global minimum point $[0, 0]^T$ is easily computed as

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}, \tag{10}$$

which is a convex nonlinear convex problem with rank defects. Simply choose the activation function $s(x) = x$ and let $k = 100$, $\lambda = 0.0001$ in the neural network (7). We obtain the following differential equations,

$$\begin{cases} \frac{dx_1}{dt} = -4x_1^3 - 2x_1 - c_1 x_1 y_1^2 - 24 x_1 y_1 z_1 - c_2 x_1 y_1 \left[(12x_1^2 + 2) y_1 + c_3 y_2 - 0.20 \right] \\ \frac{dx_2}{dt} = -4x_2^3 - c_1 x_2 y_2^2 - 24 x_2 y_2 z_2 - c_2 x_2 y_2 \left[c_3 y_1 + (12x_2^2 + 0.041) y_2 - 0.60 \right] \\ \frac{dy_1}{dt} = (-c_1 x_1^2 - 0.0004) y_1 - (12x_1^2 + 2.0) z_1 - c_3 z_2 - (c_4 x_1^2 + 200) \left[(12x_1^2 + 2.0) y_1 \right. \\ \qquad \left. + c_3 y_2 - 0.20 \right] - 0.079 y_1 - (34x_2^2 + 0.12) y_2 + 1.7 \\ \frac{dy_2}{dt} = -c_1 x_2^2 y_2 - c_3 z_1 - (12x_2^2 + 0.041) z_2 - (34x_1^2 + 5.7) y_1 - 0.079 y_2 + 0.56 \\ \qquad - (c_4 x_2^2 + 4.1) \left[c_3 y_1 + (12x_2^2 + 0.041) y_2 - 0.60 \right] \\ \frac{dz_1}{dt} = (-12x_1^2 - 2.0) y_1 - c_3 y_2 + 0.20 \\ \frac{dz_2}{dt} = -c_3 y_1 + (-12x_2^2 - 0.041) y_2 + 0.60 \end{cases} \tag{11}$$

where contants $c_1 = 0.0024, c_2 = 2400, c_3 = 0.028, c_4 = 1200$. Randomly choose initial points $x = [0.12, 0.38]^T$, the result is shown in Fig. 2. It can be seen that the neural network model successfully found the global minimum point $[0, 0]^T$. Starting from five different initial points in $[-1, 1] \times [-1, 1]$, phase portrait of the proposed neural network model is shown in Fig. 3. All trajectories asymptotically approach the global minimum point, which further verified the effectiveness of the proposed neural model.

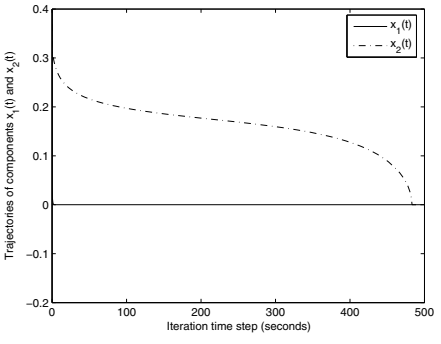


Fig. 2. Convergence of $x(t)$ to $(0, 0)$

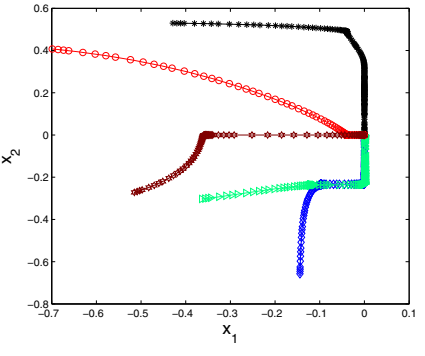


Fig. 3. Phase portrait for different initial points

5 Concluding Remarks

Singular nonlinear convex optimization problems have been traditionally studied by classical numerical methods. In this paper, a novel neural network model was established to solve such a difficult problem. Under some mild assumptions, the unconstrained nonlinear optimization problem is turned into a constrained optimization problem. By establishing the relationship between KKT points and the augmented Lagrange function, a neural network model is successfully obtained. Global analysis simulations with simple example supports the presented results.

References

1. Tank, D.W., Hopfield, J.J.: Simple neural Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit. IEEE Trans. Circuits Syst. CAS-33, 533–541 (1986)
2. Kennedy, M.P., Chua, L.O.: Neural Networks for Nonlinear Programming. IEEE Trans. Circuits Syst. 35, 554–562 (1988)
3. Bouzerdoum, A., Pattison, T.R.: Neural Network for Quadratic Optimization with Bound Constraints. IEEE Trans. Neural Networks 4, 293–304 (1993)
4. Liang, X.B., Wang, J.: A Recurrent Neural Network for Nonlinear Optimization with a Continuously Differentiable Objective Function and Bound Constraints. IEEE Trans. Neural Networks 11, 1251–1262 (2000)
5. Xia, Y.S., Wang, J.: On The Stability Of Globally Projected Dynamical Systems. J. Optim. Theory Applicat. 106, 129–150 (2000)

6. Xia, Y.S., Wang, J.: A New Neural Network for Solving Linear Programming Problems and Its Applications. IEEE Trans. Neural Networks 7, 525–529 (1996)
7. Xia, Y.S.: A New Neural Network for Solving Linear and Quadratic Programming Problems. IEEE Trans. Neural Networks 7, 1544–1547 (1996)
8. Xia, Y.S., Wang, J.: A General Methodology for Designing Globally Convergent Optimization Neural Networks. IEEE Trans. Neural Networks 9, 1331–1343 (1998)
9. Xia, Y.S.: A Recurrent Neural Network for Solving Linear Projection Equations. Neural Networks 13, 337–350 (2000)
10. Xia, Y.S.: A Dual Neural Network for Kinematic Control Of Redundant Robot Manipulators. IEEE Trans. Syst., Man, Cybern. B 31, 147–154 (2001)
11. Xia, Y.S., Leung, H., Wang, J.: A Projection Neural Network and Its application to Constrained Optimization Problems. IEEE Trans. Circuits Syst. I 49, 447–458 (2002)
12. Xia, Y.S., Wang, J.: A Generla Projection Neural Network for Solving Monotone Variational Inequality and Related Optimization Problems. IEEE Trans. Neural Networks 15, 318–328 (2004)
13. Gao, X., Liao, L.Z., Xue, W.: A Neural Network For a Class Of Convex Quadratic Minimax Problems with Constraints. IEEE Trans. Neural Netw. 15, 622–628 (2004)
14. Sun, C.Y., Feng, C.B.: Neural Networks for Nonconvex Nonlinear Programming Problems: A Switching Control Approach. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3496, pp. 694–699. Springer, Heidelberg (2005)
15. Tao, Q., Liu, X., Xue, M.S.: A Dynamic Genetic Algorithm Based on Continuous Neural Networks For a Kind of Non-Convex Optimization Problems. Appl. Math. Comput. 150, 811–820 (2004)
16. Ge, R., Xia, Z.: Solving a Type Of Modifed BFGS Algorithm with Any Rank Defects and The Local Q-Superlinear Convergence Properties, J. Computational & Applied Mathematics 22, 1–2 (2006)
17. Ge, R., Xia, Z.: A Type Of Modified BFGS Algorithm With Rank Defects And Its Global Convergence In Convex Minimization. Journal of Pure And Applied Mathematics: Advances And Applications 3, 17–35 (2010)
18. Du, X., Yang, Y., Li, M.: Further Studies on The Hestenes-Powell Augmented Lagrangian Function for Equality Constraints in Nonlinear Programming Problems. OR Transactions 10, 38–46 (2006)

An Extended TopoART Network for the Stable On-line Learning of Regression Functions

Marko Tscherepanow

Applied Informatics, Bielefeld University
Universitätsstraße 25, 33615 Bielefeld, Germany
marko@techfak.uni-bielefeld.de

Abstract. In this paper, a novel on-line regression method is presented. Due to its origins in Adaptive Resonance Theory neural networks, this method is particularly well-suited to problems requiring stable incremental learning. Its performance on five publicly available datasets is shown to be at least comparable to two established off-line methods. Furthermore, it exhibits considerable improvements in comparison to its closest supervised relative Fuzzy ARTMAP.

Keywords: Regression, On-line learning, TopoART, Adaptive Resonance Theory.

1 Introduction

For many machine learning problems, the common distinction between a training and an application phase is not reasonable (e.g., [1,2]). They rather require the gradual extension of available knowledge when the respective learning technique is already in application. This task can be fulfilled by on-line learning approaches. But in order to use on-line learning, additional problems have to be tackled. Probably the most important question is how new information can be learnt without forgetting previously gained knowledge in an uncontrolled way. This question is usually referred to as the *stability-plasticity dilemma* [3]. In order to solve it, Adaptive Resonance Theory (ART) neural networks were developed, e.g., Fuzzy ART [4] and TopoART [5].

In this paper, a regression method based on the recently published TopoART model [5] is presented. As well as being able to incrementally learn stable representations like other ART networks, TopoART is less sensitive to noise as it possesses an effective filtering mechanism. But since ART networks constitute an unsupervised learning technique, TopoART had to be extended in order to adapt it to the application field of regression.

In Section 2, an overview of regression methods, in general, and particularly related approaches is provided. Then, TopoART is briefly introduced in Section 3. Afterwards, the required extensions of TopoART are explained in Section 4. The resulting regression method is referred to as TopoART-R. It is evaluated using several datasets originating from the UCI machine learning repository [6] (see Section 5). Finally, the most important outcomes are summarised in Section 6.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 562–571, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Related Work

Regression analysis estimates a regression function f relating a set of p independent variables i_k to q dependent variables d_k:

$$\underline{d} = f(\underline{i}) \quad \text{, with } \underline{i} = [i_1, \ldots, i_p]^T \text{ and } \underline{d} = [d_1, \ldots, d_q]^T . \tag{1}$$

The models and techniques used to approximate f vary considerably; for example, a linear model can be used [7]. Although this model is only capable of reflecting linear dependencies, its parameters (slope, y-intercept) can directly be derived from observed data without the need for an explicit optimisation. In contrast, more advanced models such as support vector regression (SVR) [8] or multi-layer perceptrons (MLPs) [9] can be applied so as to model complex dependencies. But the underlying models have to be optimised by solving a quadratic optimisation problem and by gradient descent, respectively. Recently, extreme learning machines (ELMs) [10] have been proposed as a special type of MLPs possessing a single hidden layer. Here, the weights and biases of the hidden nodes are randomly assigned and the weights of the output nodes are analytically determined based on a given training set.

In recent years, several approaches to on-line SVR have been proposed [11,12]. Since new input may change the role of previously learnt data in the model, they require the complete training set to be stored. In contrast to SVR, MLPs are inherently capable of on-line learning. But the training with new input alters already-learnt representations and the network topology has to be chosen in advance. The latter problem was solved by the Cascade-Correlation (CasCor) architecture [9,13]. CasCor incrementally creates a multi-layer structure, but demands batch-learning.

As mentioned above, ART networks [4,5] constitute a solution to the stability-plasticity dilemma. They learn a set of templates (categories) which efficiently represents the underlying data distribution; new categories are incorporated, if required. Therefore, they are particularly well-suited to incremental on-line learning. ART networks can be applied to supervised learning tasks using the ARTMAP approach [14]. ARTMAP combines two ART modules, called ART_a and ART_b, by means of an associative memory (map field). While ART_a clusters $\underline{i}$, ART_b clusters $\underline{d}$. Furthermore, associations from categories of ART_a to categories of ART_b are learnt in the map field. Although, in principle, the dependent variables can be reconstructed based on the associated categories, ARTMAP cannot directly be applied as a regression method. But there exist ARTMAP variants dedicated to classification such as Default ARTMAP [15]. Default ARTMAP has a simplified structure omitting the map field and ART_b. Moreover, it enables a distributed activation during prediction, which increases the classification accuracy.

In this paper, a regression method based on TopoART is proposed. In order to increase its accuracy, a distributed activation during prediction similar to Default ARTMAP was incorporated.

3 TopoART

Like Fuzzy ART [4], TopoART [5] represents input samples by means of hyperrectangular categories. These categories as well as the associated learning mechanisms avoid catastrophic forgetting and enable the formation of stable representations. Similar to the Self-Organising Incremental Neural Network (SOINN) [16], TopoART is capable of learning the topological structure of the input data at two different levels of detail. Here, interconnected categories form arbitrarily shaped clusters. Moreover, it has been shown to be insensitive to noise as well. But TopoART requires significantly fewer parameters to be set and can learn both representational levels in parallel. Figure 1 shows the clusters resulting from training TopoART[1] with a 2-dimensional dataset comprising 20,000 samples, 10 percent of which are uniformly distributed random noise.

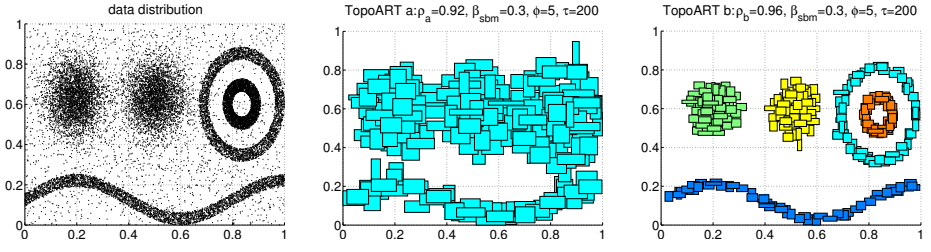


Fig. 1. Input distribution and clustering results of TopoART. After presenting each training sample of the dataset (a) to a TopoART network, it created a noise-free representation at two levels of detail. While only one cluster was formed by TopoART *a* (b), TopoART *b* distinguishes five clusters reflecting the data distribution in more detail (c). The categories associated with the same cluster share a common colour.

The two representational levels are created by two identical modules called TopoART *a* and TopoART *b*. As TopoART *a* controls which input samples are propagated to TopoART *b*, it functions as a filtering mechanism; in particular, only samples, which are enclosed by a category of TopoART *a* are propagated to TopoART *b*. In this way, noise regions are filtered effectively. Furthermore, the maximum category size is reduced from TopoART *a* to TopoART *b*. As a result, the structures represented by TopoART *b* exhibit a higher level of detail.

4 Using TopoART for Regression Analysis

Even though regression analysis constitutes a completely new application field for TopoART, its principal structure and mechanisms were directly adopted (see Fig. 2): TopoART-R consists of two modules (TopoART-R *a* and TopoART-R *b*)

[1] LibTopoART (version 0.20), available at `www.LibTopoART.eu`

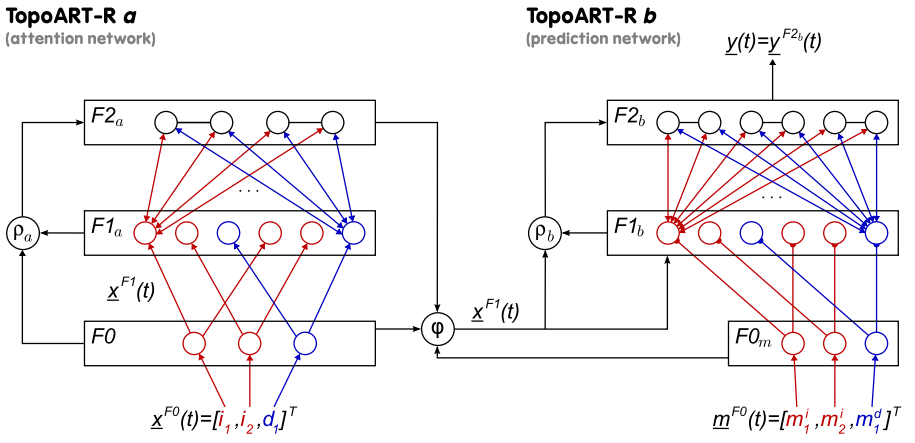


Fig. 2. Structure of TopoART-R. Like TopoART, TopoART-R encompasses two modules (TopoART-R a and TopoART-R b) sharing the input layer $F0$. But the connections of the F2 neurons can either be traced back to $\underline{i}$ or to $\underline{d}$. Furthermore, TopoART-R b has an additional input control layer ($F0_m$) that is required for prediction.

performing a clustering of the input at different levels of detail. As a consequence, the properties mentioned in Section 3 hold for the new application field as well. Nevertheless, several extensions had to be incorporated.

During training, the propagation of input to TopoART-R b depends on the activation of TopoART-R a: only input samples lying in a subspace defined by TopoART-R a reach TopoART-R b. Therefore, it is also called the 'attention network'. Predictions are provided by TopoART-R b. In order to fulfil this task, it requires the additional control layer $F0_m$.

4.1 Training TopoART-R

During training, the independent variables i_k and the dependent variables d_k are treated in the same way. For each time step t, the corresponding vectors $\underline{i}(t)$ and $\underline{d}(t)$ are concatenated and fed as input $\underline{x}^{F0}(t)$ into the TopoART-R network:

$$\underline{x}^{F0}(t) = \begin{bmatrix} \underline{i}(t) \\ \underline{d}(t) \end{bmatrix} = \big[i_1(t), \ldots, i_p(t), d_1(t), \ldots, d_q(t)\big]^T. \tag{2}$$

At the $F0$ layer, the input vectors $\underline{x}^{F0}(t)$ are encoded using complement coding:

$$\underline{x}^{F1}(t) = \big[i_1(t), \ldots, d_q(t), 1 - i_1(t), \ldots, 1 - d_q(t)\big]^T. \tag{3}$$

Due to the usage of complement coding, each element of an input vector $\underline{x}^{F0}(t)$ has to lie in the interval $[0, 1]$.

The set $\mathcal{I}$ summarises the indices of the elements of $\underline{x}^{F1}(t)$ related to $\underline{i}(t)$ and its complement, while the set $\mathcal{D}$ gives the indices for $\underline{d}(t)$ and its complement:

$$\mathcal{I} = \{1, \ldots, p, p{+}q{+}1, \ldots, 2p{+}q\}, \tag{4}$$

$$\mathcal{D} = \{p{+}1, \ldots, p{+}q, 2p{+}q{+}1, \ldots, 2p{+}2q\}. \tag{5}$$

The complement-coded input vectors $\underline{x}^{F1}(t)$ are propagated to the $F1$ layer of TopoART-R a. Then, the $F2$ nodes j of TopoART-R a are activated:

$$z_j^{F2_a}(t) = \frac{\left|\underline{x}^{F1}(t) \wedge \underline{w}_j^{F2_a}(t)\right|_1}{\alpha + \left|\underline{w}_j^{F2_a}(t)\right|_1} \quad , \text{with} \quad \alpha > 0. \tag{6}$$

Here, $|\cdot|_1$ and $\wedge$ denote the city block norm and an element-wise minimum operation, respectively. The activation $z_j^{F2}(t)$ (choice function) measures the similarity between $\underline{x}^{F1}(t)$ and the category represented by neuron j. Like with the original TopoART, the weights $\underline{w}_j^{F2_a}(t)$ span hyperrectangular categories.

The $F2$ node that has the highest activation is selected as the best-matching node bm. But it is only allowed to learn $\underline{x}^{F1}(t)$ if it fulfils the match function

$$\frac{\left|\underline{x}^{F1}(t) \wedge \underline{w}_{bm}^{F2_a}(t)\right|_1}{\left|\underline{x}^{F1}(t)\right|_1} \geq \rho_a; \tag{7}$$

i.e., if the category represented by its weights $\underline{w}_{bm}^{F2_a}(t)$ is able to enclose the presented input vector without surpassing a maximum size defined by the vigilance parameter ρ_a.

Using the original match function (7), a high variance of the dependent variables d_k could be compensated for by a low variance of the independent variables i_k. The result would be a high regression error. Therefore, the match function is independently computed for both components of the input vector $\underline{x}^{F0}(t)$:

$$\frac{\sum_k \min\left(x_k^{F1}(t), w_{bm,k}^{F2_a}(t)\right)}{\sum_k x_k^{F1}(t)} \geq \rho_a \quad , \text{for} \quad k \in \mathcal{I} \quad \text{and for} \quad k \in \mathcal{D}. \tag{8}$$

If (8) can be fulfilled, resonance of TopoART-R a occurs. Otherwise, the activation of neuron bm is reset and a new best-matching node is searched for. If no existing neuron is able to represent $\underline{x}^{F1}(t)$, a new node with $\underline{w}_{new}^{F2_a}(t{+}1){=}\underline{x}^{F1}(t)$ is incorporated.

Provided that TopoART-R a reached resonance, the weights $\underline{w}_{bm}^{F2_a}(t)$ are adapted as follows:

$$\underline{w}_{bm}^{F2_a}(t+1) = \underline{x}^{F1}(t) \wedge \underline{w}_{bm}^{F2_a}(t). \tag{9}$$

If a second-best-matching neuron sbm fulfilling (8) can be found, its weights are adapted as well:

$$\underline{w}_{sbm}^{F2_a}(t+1) = \beta_{sbm}\left(\underline{x}^{F1}(t) \wedge \underline{w}_{sbm}^{F2_a}(t)\right) + (1-\beta_{sbm})\underline{w}_{sbm}^{F2_a}(t). \tag{10}$$

This is intended to reduce the sensitivity to noise, since the growth of categories in relevant areas of the input space is intensified.

As the weights are adapted after the presentation of single input samples and TopoART-R does not rely on the processing of whole datasets in order to compute weight changes (batch learning), it is always trained on-line.

In contrast to TopoART, no edge needs to be established between node *bm* and node *sbm*, as the topological structure of the input data is not used by TopoART-R. However, TopoART-R could learn topological structures, as well, if required by future applications.

Besides its weights, each $F2$ neuron j has a counter denoted by n_j^a, which counts the number of input samples it has learnt. Every τ learning cycles, all neurons with a counter smaller than ϕ are removed. Therefore, they are called *node candidates*. After n_j^a has reached the value of ϕ, the corresponding neuron can no longer be removed; i.e., it has become a permanent node.

$\underline{x}^{F1}(t)$ is only propagated to TopoART-R b if one of the two following conditions is fulfilled:

(i) TopoART-R a is in resonance and $n_{bm}^a \geq \phi$.
(ii) The input control layer $F0_m$ is activated; i.e., $\left|\underline{m}^{F0}(t)\right|_1 > 0$.

As during training all elements of $\underline{m}^{F0}(t)$ are set to 0, only input samples which lie in one of the permanent categories of TopoART-R a are learnt by TopoART-R b. By means of this procedure, the network becomes more insensitive to noise but is still able to learn stable representations.

After input has been presented to TopoART-R b, it is activated and adapted in the same way like TopoART-R a. Just the vigilance parameter is modified:

$$\rho_b = \frac{1}{2}(\rho_a + 1). \tag{11}$$

As a result of the increased value of the vigilance parameter, TopoART-R b represents the input distribution in more detail.

4.2 Predicting with TopoART-R

In order to predict missing variables with TopoART-R, the mask vector $\underline{m}^{F0}(t)$ must be set accordingly. Consequently, TopoART-R a can be neglected, as $\left|\underline{m}^{F0}(t)\right|_1 > 0$ (see Section 4.1). The mask vector comprises the values m_k^i and m_k^d which correspond to the elements of the input vector $\underline{x}^{F0}(t)$:

$$\underline{m}^{F0}(t) = \begin{bmatrix} \underline{m}^i(t) \\ \underline{m}^d(t) \end{bmatrix} = \left[m_1^i(t), \ldots, m_p^i(t), m_1^d(t), \ldots, m_q^d(t)\right]^T. \tag{12}$$

If these mask values are set to 1, the corresponding variables are to be predicted. Hence, they cannot be given in $\underline{x}^{F0}(t)$ and the respective elements of $\underline{x}^{F0}(t)$ are ignored. Presented variables are characterised by a mask value of 0. Hence, $m_k^i = 0$ and $m_k^d = 1$ for usual regression tasks. TopoART-R can even predict based on incomplete information; if the value of an independent variable i_l is unknown, m_l^i has to be set to 1. Then, i_l is not required as input and will be predicted like the dependent variables.

Each connection of all $F2_b$ neurons can be traced back to a specific element of the input vector $\underline{x}^{F0}(t)$ and to two elements of the complement-coded input

vector $\underline{x}^{F1}(t)$ (see Fig. 2). Depending on the corresponding mask values, two disjunct sets $\mathcal{M}^0$ and $\mathcal{M}^1$ of $F1_b$ nodes are generated:

$$\mathcal{M}^0 = \left\{x, x{+}p{+}q : m_x^{F0}(t) = 0\right\}, \tag{13}$$

$$\mathcal{M}^1 = \left\{x : m_x^{F0}(t) = 1\right\}. \tag{14}$$

As the neurons of the mask layer $F0_m$ inhibit the corresponding $F1_b$ nodes (see Fig. 2), the activation of the $F2_b$ neurons is computed solely based on the non-inhibited F1 neurons summarised in $\mathcal{M}^0$. The activation function suggested for prediction with TopoART (cf., [5]) had to be adapted accordingly:

$$z_j^{F2_b}(t) = 1 - \frac{\sum_{k\in\mathcal{M}^0} \left|\min\left(x_k^{F1}(t), w_{jk}^{F2_b}(t)\right) - w_{jk}^{F2_b}(t)\right|}{\frac{1}{2}\sum_{k\in\mathcal{M}^0} 1}. \tag{15}$$

The activation $z_j^{F2_b}(t)$ computed according to (15) therefore denotes the similarity of $\underline{x}^{F1}(t)$ with $\underline{w}_j^{F2_b}(t)$ along those dimensions for which $m_x^{F0}(t){=}0$. The corresponding hyperrectangle is called a partial category.

In order to reconstruct the missing variables using a distributed activation, two cases are distinguished. Firstly, $\underline{x}^{F1}(t)$ lies inside the partial categories of one or more $F2_b$ neurons j. Then, the activation $z_j^{F2_b}(t)$ equals 1 for these neurons. Secondly, $\underline{x}^{F1}(t)$ is not enclosed by any partial category; i.e., the activation of all $F2_b$ neurons is lesser than 1.

In the first case, the missing variables are determined based on the information encoded in the partial categories: a temporary category $\underline{\tau}(t)$ is computed as the intersection of all categories that enclose $\underline{x}^{F1}(t)$. This intersection decreases in size if more neurons are involved. Thus, the more partial categories contain $\underline{x}^{F1}(t)$, the better is it represented by the network.

Since the weight vectors encode lower and upper bounds along all coordinate axes, the intersection is computed as the hyperrectangle with the respective largest lower bound and the smallest upper bound over all considered categories. Due to the usage of complement coding, this operation can be performed using the element-wise maximum operator $\vee$:

$$\underline{\tau}(t) = \bigvee_j \underline{w}_j^{F2_b}(t) \quad , \forall j : z_j^{F2_b}(t) = 1. \tag{16}$$

As $\underline{\tau}(t)$ covers all dimensions including those corresponding to the missing variables, it can be applied for computing predictions. These predictions are summarised in the output vector $\underline{y}(t)$. Its elements $y_k(t)$ are set to -1 if the corresponding variable was contained in the input vector $\underline{x}^{F0}(t)$. Otherwise, it gives a prediction which is computed as the mean of the temporary category's upper and lower bound along the k-th axis of the input space:

$$y_k(t) = \begin{cases} -1 & , \text{for } \; k \notin \mathcal{M}^1 \\ \frac{1}{2}\tau_k(t) + \frac{1}{2}\left(1 - \tau_{k+p+q}(t)\right) & , \text{for } \; k \in \mathcal{M}^1 \end{cases}. \tag{17}$$

In the second case, i.e, if no partial category contains $\underline{x}^{F1}(t)$, an intersection similar to (16) does not lead to a valid temporary category. Therefore, the temporary category is constructed as a weighted combination of the categories with the smallest distances to $\underline{x}^{F1}(t)$:

$$\underline{\tau}(t) = \frac{\sum_{j \in \mathcal{N}} \left(\frac{1}{1 - z_j^{F2_b}(t)} \cdot \underline{w}_j^{F2_b}(t) \right)}{\sum_{j \in \mathcal{N}} \frac{1}{1 - z_j^{F2_b}(t)}}. \tag{18}$$

The contribution of each node j is inversely proportional to $1 - z_j^{F2_b}(t)$; i.e., more similar categories have a higher impact. The set $\mathcal{N}$ of very similar categories is determined as follows:

$$\mathcal{N} = \{x : z_x^{F2_b}(t) \geq \mu + 1.28\sigma\}. \tag{19}$$

Here, μ and σ denote the mean and the standard deviation of $z_j^{F2_b}(t)$ over all $F2_b$ neurons. Assuming a Gaussian distribution, $\mathcal{N}$ would only contain those 10% of the neurons that have the highest activations. For computational reasons, $\mathcal{N}$ is further restricted to a maximum of 10 nodes.

5 Results

For the evaluation of TopoART-R, we chose five different datasets from the UCI machine learning repository [6]: Concrete Compressive Strength [17], Concrete Slump Test [18], Forest Fires[2] [19], and Wine Quality [20]. These datasets were selected, since they can be used with regression methods and contain real-valued attributes without missing values. For computational purposes and comparison reasons, all variables were normalised to the interval $[0, 1]$.

The performance of TopoART-R was compared to three different state-of-the-art methods: ν-SVR (with a radial basis function kernel) implemented in LIBSVM (version 3.1), CasCor, and Fuzzy ARTMAP. SVR and CasCor learn the regression function in batch mode; i.e., the training requires a complete dataset to be available. In contrast, Fuzzy ARTMAP and TopoART-R learn a sample directly after its presentation independently of other samples (on-line learning). Since Fuzzy ARTMAP learns a mapping to categories representing the dependent variables rather than a mapping to the dependent variables themselves (cf. Section 2), the centre of the ART_b category connected to the best-matching node of the map field was used as prediction.

For all regression methods, the mean squared error (MSE) was computed for each dataset using five-fold cross-validation. The most relevant parameters were determined by means of grid search.[3] The minimum MSEs reached by

[2] The integer attributes X and Y as well as the nominal attributes month and day were ignored.

[3] SVR: ν, C, and γ; CasCor: learning rate and activation function of the output nodes (logistic, arctan, tanh); Fuzzy ART: ρ, β, and β_{ab}; TopoART: ρ_a, ϕ, and β_{sbm}

each approach using the optimal parameter setting are given in Table 1. For SVR and CasCor, the respective batch learning scheme was applied. Since the number of samples contained in the datasets is rather small (e.g., 103 samples in the Concrete Slump Test dataset), the training sets were repeatedly presented to Fuzzy ART and TopoART until their weights converged. Although these methods learn on-line, they require a sufficiently high number of training steps which depends on the chosen learning rates (β, β_{ab}, and β_{sbm}).

Table 1. Minimum MSEs. The bold numbers indicate the best result for each dataset.

dataset	SVR	CasCor	Fuzzy ARTMAP	TopoART-R
Concrete Compressive Strength	**0.0054**	0.0069	0.0302	0.0119
Concrete Slump Test	0.0656	**0.0370**	0.0597	0.0475
Forest Fires	0.0034	0.0035	0.0037	**0.0032**
Wine Quality (red)	0.0161	0.0164	0.0188	**0.0143**
Wine Quality (white)	0.0122	0.0147	0.0173	**0.0105**

Table 1 shows that TopoART-R achieved the lowest MSEs for three of five datasets. Furthermore, it performed always better than Fuzzy ARTMAP, which is its closest supervised relative. Thus, TopoART-R constitutes a promising alternative to established regression methods.

6 Conclusion

In this paper, a regression method based on the unsupervised TopoART network was introduced. Due to its origins in ART networks, it is particularly suited to tasks requiring stable on-line learning. The performance of TopoART-R on standard datasets has been shown to be excellent. This is most likely a result of its noise reduction capabilities inherited from TopoART as well as the distributed activation during prediction. Finally, TopoART-R offers some properties which might be of interest for future applications: it can learn the topological structure of the presented data similar to TopoART and predict based on incomplete information if the mask vector is set appropriately. The latter property could be crucial if predictions are to be made using data from sensors with different response times.

Acknowledgements. This work was partially funded by the German Research Foundation (DFG), Excellence Cluster 277 “Cognitive Interaction Technology”.

References

1. Lee, D.H., Kim, J.J., Lee, J.J.: Online support vector regression based actor-critic method. In: Proceedings of the Annual Conference of the IEEE Industrial Electronics Society, pp. 193–198. IEEE (2010)

2. Tscherepanow, M., Jensen, N., Kummert, F.: An incremental approach to automated protein localisation. BMC Bioinformatics 9(445) (2008)
3. Grossberg, S.: Competitive learning: From interactive activation to adaptive resonance. Cognitive Science 11, 23–63 (1987)
4. Carpenter, G.A., Grossberg, S., Rosen, D.B.: Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks 4, 759–771 (1991)
5. Tscherepanow, M.: TopoART: A Topology Learning Hierarchical ART Network. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010. LNCS, vol. 6354, pp. 157–167. Springer, Heidelberg (2010)
6. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
7. Edwards, A.L.: An Introduction to Linear Regression and Correlation. W. H. Freeman and Company, San Francisco (1976)
8. Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Neural Computation 12, 1207–1245 (2000)
9. Fausett, L.: Fundamentals of Neural Networks – Architectures, Algorithms, and Applications. Prentice Hall, New Jersey (1994)
10. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: Theory and applications. Neurocomputing 70, 489–501 (2006)
11. Ma, J., Theiler, J., Perkins, S.: Accurate on-line support vector regression. Neural Computation 15, 2683–2703 (2003)
12. Martin, M.: On-line support vector machine regression. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS (LNAI), vol. 2430, pp. 173–198. Springer, Heidelberg (2002)
13. Fahlman, S.E., Lebiere, C.: The cascade-correlation learning architecture. In: Neural Information Processing Systems, vol. 2, pp. 524–532. Morgan Kaufmann, San Mateo (1989)
14. Carpenter, G.A., Grossberg, S., Reynolds, J.H.: ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. Neural Networks 4, 565–588 (1991)
15. Carpenter, G.A.: Default ARTMAP. In: Proceedings of the International Joint Conference on Neural Networks, vol. 2, pp. 1396–1401. IEEE (2003)
16. Furao, S., Hasegawa, O.: An incremental network for on-line unsupervised classification and topology learning. Neural Networks 19, 90–106 (2006)
17. Yeh, I.C.: Modeling of strength of high performance concrete using artificial neural networks. Cement and Concrete Research 28(12), 1797–1808 (1998)
18. Yeh, I.C.: Modeling slump flow of concrete using second-order regressions and artificial neural networks. Cement and Concrete Composites 29(6), 474–480 (2007)
19. Cortez, P., Morais, A.: A data mining approach to predict forest fires using meteorological data. In: Proceedings of the Portuguese Conference on Artificial Intelligence. LNAI, vol. 4874, pp. 512–523. Springer, Berlin (2007)
20. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J.: Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems 47(4), 547–553 (2009)

Introducing Reordering Algorithms to Classic Well-Known Ensembles to Improve Their Performance

Joaquín Torres-Sospedra, Carlos Hernández-Espinosa, and Mercedes Fernández-Redondo

Department of Computer Science and Engineering, Universitat Jaume I
Avda. Sos Baynat s/n, CP E-12071, Castellón, Spain
{jtorres,espinosa,redondo}@icc.uji.es

Abstract. Most of the well-known ensemble techniques use the same training algorithm and the same sequence of patterns from the learning set to adapt the trainable parameters (weights) of the neural networks in the ensemble. In this paper, we propose to replace the traditional training algorithm in which the sequence of patterns is kept unchanged during learning. With the new algorithms we want to add diversity to the ensemble and increase its accuracy by altering the sequence of patterns for each concrete network. Two new training set reordering strategies are proposed: *Static reordering* and *Dynamic reordering*. The new algorithms have been successfully tested with six different ensemble methods and the results show that reordering is a good alternative to traditional training

Keywords: Backpropagation, Ensembles of ANN, Reordering of Training set.

1 Introduction

Reviewing the literature, it can be seen that ensembles of neural networks are widely used to solve classification problems. The generalization ability of a single network can be improved with this procedure only if the networks that compose the ensemble are uncorrelated (they do not commit the same errors) [12]. Most of the ensembles tend to generate the different networks by changing the learning set: such as *Bagging* [2], *Boosting* [5,7] or *CVC* [9,13]. However, there are other alternatives, such as *DECO* [10], *EENCL* [8] among other ensembles, that modify the structure of the training algorithm.

Although the *Backpropagation* algorithm has been used to adapt the weights of the networks, it has been slightly modified in a few ensembles (i.e. a term is used in *DECO* to penalize the correlation between two consecutive networks) but there has not been introduced, in general, any constraint related to the sequence of patterns in the training set. In particular, the order in which the patterns are presented to the network during training is fixed for every epoch of the learning procedure. However, this order can be randomly changed. This

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 572–579, 2011.
© Springer-Verlag Berlin Heidelberg 2011

means that the sequence of the patterns from the training set can be exclusively set for every network in the ensemble. Moreover, the sequence can also be altered for every iteration or epoch of the *Backpropagation* algorithm when the usual on-line version is used.

For this reason, we propose the application of two new reordering methods to all the traditional ensembles when possible. Diversity may be increased if the same sequence of patterns during the training procedure is not used for all the networks. Applying reordering algorithms, two networks may not fall into the same configuration because the "path" to obtain it is different in all of them.

This paper is organized as follows. In sections 2 and 3, the theoretical background and the new learning procedures are introduced. The experimental setup is in section 4 whereas the results and their analysis are in sections 5 and 6.

2 Theoretical Background

2.1 Original Training Algorithm

The architecture chosen for the experiments is the *Multilayer Feedforward Network*(MF) and its learning procedure is described in algorithm 1.

Algorithm 1. Original Network Training$\{T$, V , $net\}$

```
Set initial weights randomly inside a small interval
for e = 1 to N_epochs do
  for i = 1 to N_patterns do
    Select pattern x_i from T, the training set
    Adjust the trainable parameters
  end for
  Calculate MSE over validation set V
  Save epoch weights and calculated MSE
end for
Select epoch with lowest validation MSE
Assign best epoch configuration to the network and save it
```

2.2 Ensemble Methods

The ensembles used in the experiments are: *Simple Ensemble*, *Cross Validation Committee version 3* (*CVCv3*) [11], *Decorrelated version 1* (*DECOv1*) [10], *Conservative Boosting* (*Conserboost*) [7] and *Evolutionary Ensemble with Negative Correlation Learning* (*EENCL*) [8].

Simple Ensemble is included because it is the easiest way to generate an ensemble. *Conserboost* and *CVCv3* have been selected because they provide the best results according to previous research we have performed [4,6,11]. We want to know if they can be improved by using the new reordering algorithms.

Finally, *DECOv1* and two versions of *EENCL* (*EENCL-LG* and *EENCL-BG*) have been selected since they report good results (see also [4,6,11]) although they are not as well-known as the other ensemble alternatives.

There are more ensembles in the literature, but we have selected these six different methods to perform our task because they are widely representative and they provide good results with the traditional learning procedure.

3 Reordering Algorithms

As mentioned in the literature, all the networks of an ensemble converge into some different configurations [3]. That is the reason why the performance of the ensemble is higher than the performance of any single network that composes the ensemble. The individual networks are "different" but we can take benefit from this difference (*diversity*) when these networks are properly combined [13].

When the sequence of patterns in the training set is the same for two classifiers, they will achieve the same final network configuration if they have a common starting point, which it is given by the weight initialization, or a common "intermediate" configuration.

For this reason, we consider that it is important to apply a reordering algorithm in order to avoid this behavior. If two networks do not use the same sequence of patterns, the probability of reaching the same (or very similar) final configuration is lower. In this paper, two alternatives to reorder the training set are proposed: *Static reordering* and *Dynamic reordering*.

3.1 Static Reordering

Static reordering, can be applied to those ensembles in which the networks are trained independently. It is called *static* because the sequence of patterns for each individual network keeps unchanged during the whole training process.

In this case, the sequence of patterns is reordered at the beginning of the training procedure of every network in the ensemble as can be seen in algorithm 2. The new training set TR^{net} is drawn at random without replacement from the training set T and with the same number of patterns included in T.

Algorithm 2. Static Network Training $\{T\ ,\ V\ ,\ net\}$

```
Generate TR^net by randomly sampling T without replacement
Set initial weights randomly
for e = 1 to N_epochs do
  for i = 1 to N_patterns do
    Select pattern x_i from TR^net
    Adjust the trainable parameters of network net
  end for
  Calculate MSE over validation set V
  Save epoch weights and calculated MSE
end for
Select epoch with lowest validation MSE
Assign best epoch configuration to the network and save it
```

This reordering method can not be applied to the ensembles in which all the networks are trained simultaneously, such as *EENCL*. According to the original references of these procedures, the same pattern is presented to all the networks of the ensemble in each iteration.

Moreover, it is senseless to apply them to *Boosting* ensembles because in those methods each network has a specific training set which is not shared with the other networks. In *Conserboost* and other *Boosting* variants, the *Static reordering* is already implicit in the design procedure. Perhaps, the reordering intrinsically applied by these ensemble methods may be part of their increase of performance with respect to *Simple Ensemble*.

Furthermore, *Static reordering* will be applied to all the ensembles based on *Cross Validation Committee* because all the training sets and the sequences of patterns are similar among them.

3.2 Dynamic Reordering

The other proposed reordering algorithm, *Dynamic reordering*, can be applied to any ensemble. It is called *dynamic* because the sequence of patterns is altered sometimes during training, concretely at the beginning of every epoch. In this case, the new training set TR_e^{net} is also drawn at random without replacement from the original training set T associated to the network or ensemble and with the same number of patterns of T. This reordering is described in algorithm 3.

Algorithm 3. Dynamic Network Training$\{T$, V , $net\}$

```
Set initial weights randomly
for e = 1 to N_epochs do
   Generate TR_e^net by randomly sampling T without replacement
   for i = 1 to N_patterns do
      Select pattern x_i from TR_e^net
      Adjust the trainable parameters
   end for
   Calculate MSE over validation set V
   Save epoch weights and calculated MSE
end for
Select epoch with lowest validation MSE
Assign best epoch configuration to the network and save it
```

As shown in the algorithmic description, two different networks will not have the same sequence of patterns in any iteration of the training algorithm because it is randomly altered at the beginning of each epoch of *Backpropagation*. Moreover, it can be adapted to any ensemble procedure.

With the proposed algorithm the "path" (as sequence of patterns used for training) from the random initial configuration to the final network configuration is completely different for all the networks of the ensemble and it is constantly changing.

4 Experimental Setup

4.1 Experiments

To test the performance of the reordering algorithms proposed in this paper, ensembles of 3, 9, 20 and 40 networks have been generated according to the six ensemble methods: *Simple Ensemble*, *Decorrelated v1*, *Cross Validation Committee v3*, *Conservative Boosting* and the two implementations of *Evolutionary Ensembles with Negative Correlation Learning*. To train the individual networks, we have used the traditional training procedure and the proposed reordering algorithms, *Static Reordering* and *Dynamic Reordering*.

The success of the new reordering algorithms is shown by comparing the general performance (*mean Percentage of Error Reduction* and *Two-tailed Student's T-Test for Paired Samples*) of the ensembles with the traditional training. These measurements are highly detailed in [11].

Finally, the experiments have been repeated twenty times in every database with different partitions in training, validation an test sets. This procedure has been done in order to get a mean performance of the ensemble and its error calculated by standard error theory.

4.2 Description of the Databases

The following problems from the *UCI repository* [1] have been used to test the performance of the methods: *Abalone*, *Annealing*, *Arrhythmia*, *Australian Credit Approval*, *Balance Scale*, *Blood Transfusion Service Center*, *BUPA liver disorders*, *Congressional Voting Records*, *Contraceptive Method Choice*, *Cylinder Bands*, *Dermatology*, *Ecoli*, *Glass Identification*, *Haberman's Survival Data*, *Heart Disease*, *Image segmentation*, *Ionosphere Database*, *Mammographic Mass*, *Mushroom*, *Optical Rec. of Handwritten Digits*, *Page Blocks Classification*, *Pima Indians Diabetes*, *Solar Flares*, *Spambase*, *Statlog - German Credit Data*, *Statlog - Vehicle Silhouettes*, *The Monk's Problem 1 and 2*, *Vowel Database*, *Waveform Database Generator v1 and v2*, *Wisconsin Breast Cancer* and *Yeast*.

Due to the lack of space, the training parameters and the specific ensemble parameters (*DECOv1* and *ENNCL*) have not been included but they are publicly available in [11].

5 Results

Table 1 shows the mean PER for each ensemble and case (size). Moreover, a resume of the statistical test are included with special symbols. These symbols mean: (1 - $\bullet$) the new "reordered" ensemble is better than the traditional alternative and their differences are statistically significant ($\alpha \leq 5\%$), (2 - $\circ$) the new ensemble is better than the traditional alternative but their differences are not statistically significant and (3 - $\triangledown$) the new ensemble is worse than the traditional alternative but their differences are not statistically significant.

Table 1. Mean *Percentage of Error Reduction*

ensemble	*reordering*	*3-Net*	*9-Net*	*20-Net*	*40-Net*
SE	*no reordering*	5.6	9.2	10.8	11.2
SE	*Static*	9.2 •	11.4 •	13.2 •	13.2 •
SE	*Dynamic*	9.4 •	12 •	13.1 •	13.4 •
CVCv3	*no reordering*	8.1	12.7	14.7	15.3
CVCv3	*Static*	11.7 •	15.5 •	16.7 •	17.5 •
CVCv3	*Dynamic*	12.2 •	15.4 •	15.7 •	16.3 •
DECOv1	*no reordering*	10.4	13.2	14.8	14.9
DECOv1	*Static*	8.7 ▽	13 ○	15 •	15.4 •
DECOv1	*Dynamic*	10.5 ○	13.1 ○	13.9 ○	14.9 •
Conserboost	*no reordering*	5.9	12.4	14.5	16.1
Conserboost	*Dynamic*	6.6 ○	12.8 ○	15.4 ○	17.2 ○
EENCL-LG	*no reordering*	2.5	0.5	3.6	4.9
EENCL-LG	*Dynamic*	5.2 ○	3.9 ○	3.9 ○	4.7 ○
EENCL-BG	*no reordering*	4.9	4	4.9	8.2
EENCL-BG	*Dynamic*	9.8 •	7 ○	6.5 •	7.3 ▽

Firstly, the new reordering procedures improve the original *Simple Ensemble* in all the cases. Moreover, the differences are statistically significant as the symbol • denotes. In general, *Dynamic reordering* is better than *Static reordering*.

Secondly, similar conclusions can be reached from the results of *CVCv3* because the original version has been statistically improved by the two reordering algorithms. However, *Static reordering* is, generally, a better choice.

For *DECOv1*, the most suitable reordering algorithm depends on the ensemble size. For small ensembles (3 and 9 networks), *Dynamic reordering* is a better choice. For medium and high sized ensembles (20 and 40 networks), *Static reordering* provides the best overall results for *DECOv1* and the differences are statistically significant with respect to the traditional *DECOv1*.

In the case of *Conserboost*, *Dynamic reordering* provides better mean PER than the traditional ensemble. Their difference increases as new networks are added to the ensemble. However the results are not statistically significant. Maybe this behaviour is due to the intrinsic *Static reordering* of *Conserboost*.

Finally, *Dynamic reordering* improves *EENCL-LG* and *EENCL-BG* for 3 to 20 nets. The difference in *mean* PER is specially high for 3 nets. The differences are statistically significant in two cases, *EENCL-BG* with 3 and 20 nets.

6 Analysis of the Results

Firstly, the new reordering algorithms provide the best results in 87.5% of the cases. Concretely, *Dynamic reordering* is the best training procedure in 15 of 24 cases (62.5%) and *Static reordering* provides the best results in 6 of 24 cases (25%). We consider that reordering is an alternative which should be seriously considered.

Moreover, the *MultiTest* algorithm [14] has been used to rank (with *Pairwise Statistical Tests*) the traditional and proposed training procedures in table 2.

Table 2. Ranks according to *MultiTest* (I)

Rank for 3 nets

#	method	reordering	PER
1^{st}	*CVCv3*	*Dynamic*	12.2
2^{nd}	*CVCv3*	*Static*	11.7
3^{rd}	*DECOv1*	*Dynamic*	10.5
4^{th}	*DECOv1*	*no reordering*	10.4
5^{th}	*EENCL-BG*	*Dynamic*	9.8
6^{th}	*SE*	*Dynamic*	9.4
7^{th}	*SE*	*Static*	9.2
8^{th}	*DECOv1*	*Static*	8.7
9^{th}	*CVCv3*	*no reordering*	8.1
10^{th}	*SE*	*no reordering*	5.6
11^{th}	*EENCL-LG*	*Dynamic*	5.2
12^{th}	*EENCL-BG*	*no reordering*	4.8
13^{th}	*Conserboost*	*Dynamic*	6.5
14^{th}	*Conserboost*	*no reordering*	5.9
15^{th}	*EENCL-LG*	*no reordering*	2.5

Rank for 9 nets

#	method	reordering	PER
1^{st}	*CVCv3*	*Static*	15.6
2^{nd}	*CVCv3*	*Dynamic*	15.3
3^{rd}	*DECOv1*	*no reordering*	13.2
4^{th}	*DECOv1*	*Dynamic*	13.1
5^{th}	*DECOv1*	*Static*	13
6^{th}	*Conserboost*	*Dynamic*	12.8
7^{th}	*CVCv3*	*no reordering*	12.7
8^{th}	*Conserboost*	*no reordering*	12.4
9^{th}	*SE*	*Dynamic*	12
10^{th}	*SE*	*Static*	11.4
11^{th}	*SE*	*no reordering*	9.2
12^{th}	*EENCL-BG*	*Dynamic*	6.9
13^{th}	*EENCL-BG*	*no reordering*	4
14^{th}	*EENCL-LG*	*Dynamic*	3.9
15^{th}	*EENCL-LG*	*no reordering*	0.5

Rank for 20 nets

#	method	reordering	PER
1^{st}	*CVCv3*	*Static*	16.7
2^{nd}	*CVCv3*	*Dynamic*	15.7
3^{rd}	*Conserboost*	*Dynamic*	15.4
4^{th}	*DECOv1*	*Static*	15
5^{th}	*CVCv3*	*no reordering*	14.7
6^{th}	*Conserboost*	*no reordering*	14.6
7^{th}	*DECOv1*	*Dynamic*	13.9
8^{th}	*DECOv1*	*no reordering*	14.8
9^{th}	*SE*	*Static*	13.2
10^{th}	*SE*	*Dynamic*	13.1
11^{th}	*SE*	*no reordering*	10.8
12^{th}	*EENCL-BG*	*Dynamic*	6.5
13^{th}	*EENCL-BG*	*no reordering*	4.9
14^{th}	*EENCL-LG*	*Dynamic*	3.9
15^{th}	*EENCL-LG*	*no reordering*	3.6

Rank for 40 nets

#	method	reordering	PER
1^{st}	*CVCv3*	*Static*	17.5
2^{nd}	*Conserboost*	*Dynamic*	17.2
3^{rd}	*CVCv3*	*Dynamic*	16.3
4^{th}	*Conserboost*	*no reordering*	16.1
5^{th}	*DECOv1*	*Static*	15.4
6^{th}	*CVCv3*	*no reordering*	15.3
7^{th}	*DECOv1*	*Dynamic*	14.9
8^{th}	*DECOv1*	*no reordering*	14.9
9^{th}	*SE*	*Dynamic*	13.4
10^{th}	*SE*	*Static*	13.2
11^{th}	*SE*	*no reordering*	11.2
12^{th}	*EENCL-BG*	*no reordering*	8.2
13^{th}	*EENCL-BG*	*Dynamic*	7.3
14^{th}	*EENCL-LG*	*Dynamic*	4.7
15^{th}	*EENCL-LG*	*no reordering*	4.9

The new proposed reordering algorithms improve the traditional training procedure according to the results derived from *MultiTest*. There are only three cases of twenty-four possible cases (12.5% of cases) in which traditional training is ranked over a reordered alternative. These cases correspond to low-medium sized ensembles (3 and 9 networks) with *DECOv1* and high sized ensembles (40 networks) with *EENCL-BG*. However, in *DECOv1* with 3 nets, the traditional training is ranked below the other reordering algorithm so the traditional learning is ranked over all the new reordering alternatives only in two cases.

Moreover, the reordered versions of *CVCv3* provides the best overall results for all the sizes. Furthermore, the reordered versions of *DECOv1* and *Conserboost* also provide good results for low-medium sized ensembles (3 and 9 networks) and for medium-high sized ensembles (20 and 40 networks) respectively.

7 Conclusions

Two new reordering algorithms (*Static reordering* and *Dynamic reordering*) have been proposed to train ensembles of neural networks in this paper.

The traditional learning procedure and new reordering algorithms have been tested with six ensembles and 33 datasets. A deep analysis has been performed using the *mean PER*, the statistical *t*-test and *MultiTest*. According to these measurements, the traditional learning procedure was generally outperformed by the new reordering algorithms. Moreover, the improvements with respect to traditional training were statistically significant in half of the cases.

According to the results, the traditional learning procedure was improved by a new reordering algorithm in 91.66% of the cases. Moreover, the best overall accuracy for each individual ensemble method is also obtained if a reordering algorithm is used to train the networks. For this reason, we can conclude by remarking that the performance of traditional ensembles can be increased by altering the order of the sequence of patterns used to train the networks.

References

1. Asuncion, A., Newman, D.: UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences (2007)
2. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
3. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
4. Fernández-Redondo, M., Hernández-Espinosa, C., Torres-Sospedra, J.: Multilayer Feedforward Ensembles for Classification Problems. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) ICONIP 2004. LNCS, vol. 3316, pp. 744–749. Springer, Heidelberg (2004)
5. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning, pp. 148–156 (1996)
6. Hernández-Espinosa, C., Torres-Sospedra, J., Fernández-Redondo, M.: New experiments on ensembles of multilayer feedforward for classification problems. In: Proceedings of IJCNN 2005, pp. 1120–1124 (2005)
7. Kuncheva, L.I., Whitaker, C.J.: Using Diversity with Three Variants of Boosting: Aggressive, Conservative and Inverse. In: Roli, F., Kittler, J. (eds.) MCS 2002. LNCS, vol. 2364, pp. 81–90. Springer, Heidelberg (2002)
8. Liu, Y., Yao, X., Higuchi, T.: Evolutionary ensembles with negative correlation learning. IEEE Transactions on Evolutionary Computation 4(4), 380–387 (2000)
9. Parmanto, B., Munro, P.W., Doyle, H.R.: Improving committee diagnosis with resampling techniques. In: Advances in Neural Information Processing Systems, pp. 882–888 (1996)
10. Rosen, B.E.: Ensemble learning using decorrelated neural networks. Connection Science 8(3-4), 373–384 (1996)
11. Torres-Sospedra, J.: Ensembles of Artificial Neural Networks: Analysis and Development of Design Methods. Ph.D. thesis, Department of Computer Science and Engineering, Universitat Jaume I (2011)
12. Tumer, K., Ghosh, J.: Error correlation and error reduction in ensemble classifiers. Connection Science 8(3-4), 385–403 (1996)
13. Verikas, A., Lipnickas, A., Malmqvist, K., Bacauskiene, M., Gelzinis, A.: Soft combination of neural classifiers: A comparative study. Pattern Recognition Letters 20(4), 429–444 (1999)
14. Yildiz, O.T., Alpaydin, E.: Ordering and finding the best of k>2 supervised learning algorithms. IEEE T. Pattern. Anal. 28(3), 392–402 (2006)

Improving Boosting Methods by Generating Specific Training and Validation Sets

Joaquín Torres-Sospedra, Carlos Hernández-Espinosa,
and Mercedes Fernández-Redondo

Department of Computer Science and Engineering, Universitat Jaume I
Avda. Sos Baynat s/n, CP E-12071, Castellón, Spain
{jtorres,espinosa,redondo}@icc.uji.es

Abstract. In previous researches it can been seen that *Bagging*, *Boosting* and *Cross-Validation Committee* can provide good performance separately. In this paper, *Boosting* methods are mixed with *Bagging* and *Cross-Validation Committee* in order to generate accurate ensembles and take benefit from all these alternatives. In this way, the networks are trained according to the boosting methods but the specific training and validation set are generated according to *Bagging* or *Cross-Validation*. The results show that the proposed methodologies *BagBoosting* and *Cross-Validated Boosting* outperform the original *Boosting* ensembles.

Keywords: Ensembles of ANN, Specific sets, Boosting alternatives.

1 Introduction

Ensembles of neural networks are commonly applied in the literature in order to generate classifiers with good performance. This "approach" is used because it outperforms classifiers based on a lonely network. According to [10,13], the networks have to be "different" (with high level of diversity) in order to take benefit from them and obtain a good global performance. It is clear that if an ensemble is composed by similar networks, its performance will be also similar to the performance of any single network that composed the ensemble. The goal of an ensemble method is to obtain good individual but also different networks.

Although there are some methods to build an ensemble of neural networks: *Bagging* [3], *Boosting* [5,7,8] and *Cross-Validation Committee* [9,14] are well-known and provide good results according to previous researches [4,6].

In this paper, two new methodologies are proposed: *BagBoosting* and *Cross-Validated Boosting*. On the one hand, *Boosting* and *Cross-Validation Committee* are mixed into a single procedure to take benefit from both approaches. On the other hand, *Boosting* and *Bagging* are used together in order to improve the accuracy of the ensembles. *Adaboost*, *Aveboost* and *Conserboost* have been used as *Boosting* methods to test the performance of the new methodologies.

This paper is organized as follows. In Section 2, the leaning process of a neural network is briefly analyzed. Moreover, some *Boosting* alternatives are

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 580–587, 2011.
© Springer-Verlag Berlin Heidelberg 2011

reviewed. In section 3, the proposed methodologies, *BagBoosting* and *Cross-Validated Boosting*, are described. The experimental setup is shown in section 4 whereas the results and their analysis are in subsection 5.

2 Theoretical Background

2.1 Learning Process and Stopping Criteria

The *Multilayer Feedforward Network* (*MF network*) is the network architecture chosen for the experiments. According to [2,11], this kind of networks can approximate any function with a defined precision. Each network is trained using patterns from the training set for a predefined number of iterations (epochs). Moreover, the *Mean Squared Error* is calculated using patterns from the validation set for each epoch. The final network configuration corresponds to the weight values of the epoch with lowest error on the validation set.

2.2 Description of Boosting Methods

Adaptive Boosting. *Adaptive Boosting*, henceforth *Adaboost*, is an important ensemble proposed by Freund and Schaphire in [5]. *Adaboost* generates a sequence of networks in which the successive networks are overfitted with hard to learn patterns. A sampling distribution, *Dist*, is used to randomly create the specific training set of each successive network. This distribution is updated after training each network. In the update, the probability of selecting a pattern increases if the already trained network does not classify correctly the pattern, whereas it decreases if the pattern is correctly classified. Finally, *Adaboost* uses a specific model to combine the output provided by the networks of the ensemble.

Averaged Boosting. Oza proposed this ensemble based on *Adaboost* in [8] called *Averaged Boosting*, henceforth *Aveboost*. In *Aveboost*, the sampling distribution related to a neural network depends on three factors: 1) The sampling distribution of the previous network, 2) The equation used to update the distribution on *Adaboost* and 3) The number of networks previously trained. Moreover, the basic structure and the specific combiner are keep unchanged with respect to the *Adaboost* algorithm.

Conservative Boosting. *Conservative Boosting*, henceforth *Conserboost*, is the one of three boosting methods proposed and analyzed by Kuncheva in [7]. The description of this ensemble corresponds to the algorithmic description of *Adaboost* and the only difference between them is the equation applied to update the sampling distribution which, in this case, is softer because the probabilities are updated only if the pattern is not correctly classified. Finally, the reinitialization of the sampling distribution is allowed in this method when the value of the error of the trained network is not in the proper range. If it occurs, all the patterns will have the same probability of being selected for the specific training set in the next network.

3 Generating New Specific Sets for Boosting Methods

3.1 Specific Sets and Boosting

In the experiments the original datasets, described in section 4.2, have been divided into three subsets: Training for adapting the weights (T), Validation for selecting the final network configuration (V) and test for obtaining the performance of the network (TS). The *Learning set* is the union of T and V.

Although *Bagging* and *Cross-Validation Committee* are ensemble models, they can be used to generate specific and different training and validation sets from the original learning set. *Bagging* can be applied to generate specific training and validation sets randomly. *Cross-Validation Committee* can be used to randomly split the learning set into some subsets of the same size. Then the specific sets are generated by choosing specific subsets from them.

In the proposed methodologies, *BagBoosting* and *Cross-Validated Boosting*, the generation of the specific training and validation sets is divided into two steps. In the first one, the different training and validation sets are generated as in *Bagging* or *Cross-Validation*, T^{net} and V^{net}. Then, the training set used to train the network, T', is generated by sampling patterns from L with the sampling distribution, but the chosen pattern must be also in T^{net}.

3.2 Bagging as Set Generator

Bagging is an ensemble model but it can also be used to generate specific training and validation sets. In this paper, it is introduced to randomly generate the specific training set (T^{net}) for each network by randomly sampling patterns with replacement from the original learning set. The number of patterns of T^{net} doubles (factor n=2) the original set (T) as suggested in [13]. However, we will also generate specific sets 1.5 times greater than the original (n=1.5).

Once the specific training set is generated, T^{net}, the specific validation set, V^{net} is created. The patterns from the original learning set, L, which are not in the specific training set, T^{net} are chosen as patterns of the specific validation set, V^{net}. It means that the patterns of the specific validation set are those patterns which have not been sampled from L to generate T^{net}. A requirement of the training and validation sets is that their intersection must be the empty set.

In *BagBoosting*, *Boosting* methods are modified according to Algorithm 1.

3.3 Cross Validation Committee as Set Generator

Cross-Validation Committee can also be used to generate the specific training and validation sets. In this case, the original learning set is divided into N_{sets} subsets and the specific training and validation sets are generated from them.

Algorithm 1 can also be used to generate the new boosting methods according to *Cross-Validated Boosting*. However, it should be adapted because *Cross-Validated Boosting* and *BagBoosting* differ on how the specific sets are generated. So the final *Cross-Validated Boosting* algorithm should replace the statements

Algorithm 1. BagBoosting $\{T, V, N_{networks}\}$

Initialize Sampling Distribution $Dist$:$Dist_x^1 = 1/N_{patterns}\ \forall x \in L$
for $net = 1$ to $N_{networks}$ **do**
  Randomly Create T^{net} by sampling from L with replacement
  Generate V^{net} with the patterns from L which are not in T^{net}
  Create $T^{'}$ sampling patterns from L which are in T^{net} using $Dist^{net}$
  MF Network Training T' , V^{net}
  Update sampling distribution with a particular *boosting* method
end for

related to the procedure used to generate these sets according to the equations introduced below.

In this paper, two versions of *Cross-Validation* will be used to generate the sets: *CVCv2* and *CVCv3*.

In the first version, *CVCv2*, the number of subsets corresponds to the number of networks in the ensemble. The training set of a network, T^{net}, and its validation set, V^{net}, are given by the following equation:

$$T^{net} = \bigcup_{\substack{i=1 \\ i \neq net}}^{N_{networks}} L_i \qquad V^{net} = L_{net} \tag{1}$$

In the second version, N_{sets} has been set to 10 in order to keep the training and validation sets similar to their original sizes. In this case, the training and validation sets are given by the following equation:

$$T^{net} = \bigcup_{\substack{i=1 \\ i \neq index_{net,1} \\ i \neq index_{net,2}}}^{N_{sets}} L_i \qquad V^{net} = L_{index_{net,1}} \cup L_{index_{net,2}} \tag{2}$$

Where the indexes related to a neural network, $index_{net,1}$ and $index_{net,2}$, are randomly set with the constraint that the different networks have different training and validation sets.

4 Experimental Setup

4.1 Experiments

To test the performance of all the ensemble alternatives proposed in this paper, ensembles of 3, 9, 20 and 40 networks have been built according to the *BagBoosting* and *Cross-Validated Boosting* by using the three *Boosting methods* previously described. *Output Average* and the corresponding *Boosting Combiner* are both applied to combine these ensembles since they are the combiners originally applied in *Bagging*, *Cross-Validated Committee* and *Boosting*.

Moreover, ensembles of the same sizes (3, 9, 20 and 40 networks) have been trained with the traditional *Boosting methods*. For these ensembles, only the combiners specified in the original references are used.

Finally, the experiments have been repeated ten times in every database with different partitions in the training, validation an test sets. This procedure has been done in order to get a mean performance of the ensemble and its error calculated by standard error theory.

4.2 Description of the Databases

The following problems from the *UCI repository of machine learning databases* [1] have been used to test the performance of the methods: *Arrhythmia*, *Balance Scale*, *Cylinder Bands*, *BUPA liver disorders*, *Australian Credit Approval*, *Dermatology*, *Ecoli*, *Solar Flares*, *Glass Identification*, *Heart Disease*, *Image segmentation*, *Ionosphere Database*, *The Monk's Problem 1 and 2*, *Pima Indians Diabetes*, *Haberman's Survival Data*, *Congressional Voting Records*, *Vowel Database* and *Wisconsin Breast Cancer*. The optimal training parameters for these datasets have not been included due to the lack of space but they are publicly available in a Ph.D. thesis [12].

5 Results and Discussion

5.1 General Measurements

To perform an exhaustive comparison, the *mean Percentage of Error Reduction across all databases with respect to the Single Network* (*mean PER*) has been calculated. A PER value of 0% means that there is no improvement in the percentage of correcly classfied patterns by the use of the ensemble method with respect to a single network whereas a positive value means that the ensembles is better than the single netowrk. A value of 100% means that the error have been totally reduced. Moreover, a negative value means that a single network performs better than the ensemble. The PER value is given by eq. 3.

$$PER = 100 \cdot \frac{Perf_{Ensemble} - Perf_{Singlenet}}{100 - Perf_{SingleNet}} \tag{3}$$

Where $Perf_{singlenet}$ and $Perf_{ensemble}$ correspond to the percentage of patterns correctly classified by the single net and the ensemble respectively.

5.2 General Results

The results of the original alternatives and the new *Boosting* ensembles (*Adaboost*, *Conserboost* and *Aveboost*) using *Bagging* as partitioning procedure are introduced in table 1. For the case of the new ensembles, they have been tested with two values of the factor n (1.5 and 2). The combiners *Output Average* and *Boosting Combiner* are denoted with *-Ave* and *-Bst* in the table.

Table 1. *Mean PER - BagBoosting* methods

ensemble	*n*	*3-Net*	*9-Net*	*20-Net*	*40-Net*
Adaboost	-	15.40	19.50	22.96	24.54
BagAdaboost-Ave	1.5	18.11	23.87	24.72	22.15
BagAdaboost-Bst	1.5	16.17	23.51	24.87	25.22
BagAdaboost-Ave	2	20.09	21.77	23.15	23.68
BagAdaboost-Bst	2	18.36	22.79	25.85	24.98
Aveboost	-	18.26	26.11	27.12	26.53
BagAveboost-Ave	1.5	23.13	28.43	28.26	29.47
BagAveboost-Bst	1.5	19.63	28.35	28.97	30.26
BagAveboost-Ave	2	25.30	29.95	31.37	29.77
BagAveboost-Bst	2	19.63	28.53	31.56	29.30
Conserboost	-	19.72	25.63	26.62	27.84
BagConserboost-Ave	1.5	22.61	28.01	28.99	30.49
BagConserboost-Bst	1.5	20.08	27.17	29.65	31.38
BagConserboost-Ave	2	26.12	28.66	29.13	30.46
BagConserboost-Bst	2	23.42	27.45	30.12	31.25

Firstly, it seems that *Output Average* is a better combiner for the new ensembles when the number of networks is low-medium (in general) and *Boosting Combiner* is a better choice for high sized ensembles (specially for 40 networks).

Secondly, the best results provided by the new ensembles improve the original ensemble in all the cases. Moreover, *BagConserboost* and *BagAveboost* provide better results with 9 networks than the best overall results provided by the original methods. Some critical applications can take benefit from this fact.

Thirdly, it is less clear to decide which value fits better for the factor n because it depends on the ensemble size and combiner applied. However, the best results for the two *Boosting* methods are provided when n is equal to 2.

Fourthly, as in the original ensembles the new ensembles based on *Adaboost* seem to report the lowest performance. Ensembles based on *Aveboost* tends to be better than the ensembles based *Conserboost* except for high sized ensembles. The best overall performance is provided by *BagAveboost* (31.56%).

Then, table 2 shows the *mean PER* for the new ensembles generated with *CVC* and *Boosting* (*Cross-Validated Boosting*). Firstly, the results provided by the new ensembles improve their original boosting alternatives. In this way, the successive improvements of *Adaboost* have been outperformed. But, in the case of *Adaboost* the cross-validated variants are only better for the case of 3 and 9 networks in the ensemble.

Secondly, *Output Average* has a good performance for any ensemble, in general, whereas *Boosting Combiner* fits better on medium and high sized ensembles. However, *Boosting Combiner* report low results for low ensembles. Therefore, we can conclude that *Output Average* is the best combiner for the new methods based on *Cross-Validated Boosting* in a general way.

Thirdly, the new proposed methods based on *CVCv3* provide, in general, better results than the ones proposed and based on *CVCv2*. But, there are a few cases (specially in *Adaboost* and low-medium sized ensembles) in which the methods based on *CVC2* should also be seriously considered.

Table 2. *Mean PER - Cross-Validated Boosting* methods

ensemble	*3-Net*	*9-Net*	*20-Net*	*40-Net*
Adaboost	15.40	19.50	22.96	24.54
CVCv2Adaboost-Ave	18.3	22.7	21	9.8
CVCv2Adaboost-Bst	12.3	23.4	21.6	20.8
CVCv3Adaboost-Ave	16.7	21.6	19.8	20.4
CVCv3Adaboost-Bst	11.8	20.4	21.6	22.6
Aveboost	18.26	26.11	27.12	26.53
CVCv2Aveboost-Ave	19.4	27.3	29.3	26.5
CVCv2Aveboost-Bst	15.6	27.6	30	27
CVCv3Aveboost-Ave	21.3	27.6	29.9	30
CVCv3Aveboost-Bst	16.4	26.5	28.5	28.8
Conserboost	19.72	25.63	26.62	27.84
CVCv2Conserboost-Ave	22.6	27.6	29.1	28.1
CVCv2Conserboost-Bst	16.9	26.5	29.6	27.9
CVCv3Conserboost-Ave	22.4	28.4	30.1	30.7
CVCv3Conserboost-Bst	17.2	27.5	30.2	31.3

5.3 Analysis of the Results

First of all, the traditional and new ensemble methods have been compared according to the mean PER across all databases. The original boosting methods have been improved by applying *Bagging* and *Cross-Validation* to them in order to generate specific training and validation sets for each network.

Secondly, the new ensembles provide the highest results for each boosting ensemble when n is set to 2 in almost 60% of cases. However, the most appropriate value for n depends on the combiner and ensemble.

Thirdly, the methods based on *Cross-Validated Boosting* perform better when *CVCv3* is used to generate the specifics sets T^{net} and V^{net}. There are only a few cases, specially in low and medium sized ensembles (3 and 9 networks), in which *CVCv2* can be more suitable. So, *CVCv2* is not recommended in general due to its inherited problems derived from the fact that the size of the training set depends on the number of networks in the ensemble.

Fourthly, *Output Average* should be strongly considered to combine the networks for the proposed ensembles, specially in low and medium sized ensembles (3 to 20 networks). The specific boosting combiners shown in this paper are a simple weighted average based on the ensemble error ϵ. In the case of low and medium sized ensembles, maybe, there are not enough networks in order to get an optimal weighted average. Both combiners provide good results, depending on the ensemble method, for high sized ensembles (40 networks).

Finally, the best results provided by *BagBoosting* are better than the best results of *Cross-Validated Boosting* for the three boosting ensembles and four ensemble sizes. There is only one case of twelve possible cases in which *Cross-Validated Boosting* is better than *Bagging* as set generator (ensembles of 40 networks based on *Conserboost*). Moreover, the highest overall performance is provided by *BagAveboost* and *20* networks (*mean PER* = 31.56%).

6 Conclusions

In this paper, *Boosting methods* have been successfully fused with *Bagging* and *Cross-Validation Committee*. In general, the *Boosting* ensembles have been improved by using specific training and validation sets.

According to the research performed, *BagBoosting* improved the results of the original *Boosting* methods. In 60% of the cases the best results were obtained when n was set to 2, in the other cases n equal to 1.5 provided slightly better results. Moreover, *Cross-Validated Boosting* also improved the results provided by the traditional boosting alternatives, specially if *CVCv3* and the *Output Average* combiner were applied.

Finally, *BagBoosting* provides better results than *Cross-Validated Boosting* in all cases except one. In general, *BagBoosting* should be considered a better alternative to generate ensembles despite *Cross-Validated Boosting* also provides better results than the traditional *Boosting* methods.

References

1. Asuncion, A., Newman, D.: UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences (2007)
2. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Inc., New York (1995)
3. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
4. Fernández-Redondo, M., Hernández-Espinosa, C., Torres-Sospedra, J.: Multilayer feedforward ensembles for classification problems. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) ICONIP 2004. LNCS, vol. 3316, pp. 744–749. Springer, Heidelberg (2004)
5. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning, pp. 148–156 (1996)
6. Hernández-Espinosa, C., Torres-Sospedra, J., Fernández-Redondo, M.: New experiments on ensembles of multilayer feedforward for classification problems. In: Proceedings of IJCNN 2005, pp. 1120–1124 (2005)
7. Kuncheva, L.I., Whitaker, C.J.: Using diversity with three variants of boosting: Aggressive, conservative, and inverse. In: Roli, F., Kittler, J. (eds.) MCS 2002. LNCS, vol. 2364, pp. 81–90. Springer, Heidelberg (2002)
8. Oza, N.C.: Boosting with averaged weight vectors. In: Windeatt, T., Roli, F. (eds.) MCS 2003. LNCS, vol. 2709, pp. 15–24. Springer, Heidelberg (2003)
9. Parmanto, B., Munro, P.W., Doyle, H.R.: Improving committee diagnosis with resampling techniques. In: Advances in Neural Information Processing Systems, pp. 882–888 (1996)
10. Raviv, Y., Intratorr, N.: Bootstrapping with noise: An effective regularization technique. Connection Science 8, 356–372 (1996)
11. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge (1996)
12. Torres-Sospedra, J.: Ensembles of Artificial Neural Networks: Analysis and Development of Design Methods. Ph.D Thesis, Universitat Jaume I (2011)
13. Tumer, K., Ghosh, J.: Error correlation and error reduction in ensemble classifiers. Connection Science 8(3-4), 385–403 (1996)
14. Verikas, A., Lipnickas, A., Malmqvist, K., Bacauskiene, M., Gelzinis, A.: Soft combination of neural classifiers: A comparative study. Pattern Recognition Letters 20(4), 429–444 (1999)

Using *Bagging* and *Cross-Validation* to Improve Ensembles Based on Penalty Terms

Joaquín Torres-Sospedra, Carlos Hernández-Espinosa,
and Mercedes Fernández-Redondo

Department of Computer Science and Engineering, Universitat Jaume I
Avda. Sos Baynat s/n, CP E-12071, Castellón, Spain
{jtorres,espinosa,redondo}@icc.uji.es

Abstract. *Decorrelated* and *CELS* are two ensembles that modify the learning procedure to increase the diversity among the networks of the ensemble. Although they provide good performance according to previous comparatives, they are not as well known as other alternatives, such as *Bagging* and *Boosting*, which modify the learning set in order to obtain classifiers with high performance. In this paper, two different procedures are introduced to *Decorrelated* and *CELS* in order to modify the learning set of each individual network and improve their accuracy. The results show that these two ensembles are improved by using the two proposed methodologies as specific set generators.

Keywords: Ensembles with Penalty Terms, Specific Sets, Bagging, CVC.

1 Introduction

One technique used to generate classifiers consists in training a set of different of neural networks (ensemble). This procedure increases the generalization capability when the networks are not correlated according to the literature [11].

Although there are alternatives to generate ensembles: *Bagging* [2], *Boosting* [4] and *Cross-Validation Committee* [7] are well-known and provide good performance [3,5]. These ensembles modify the learning set to improve the accuracy of the ensemble. However, there are other ensembles such as *Deco* and *CELS* which also provides good results [3,5] but they are less used in the literature.

In this paper, two procedures to generate specific learning sets for *Decorrelated* and *CELS* are introduced. In the first one, *Bagging* is used to randomly generate the specific training and validation sets for training each network of the ensemble. In the second one, an advanced version of *Cross-Validation Committee* is used to perform the partitioning task. The second partition procedure was successfully applied to *Adaboost* in [9] so we consider that *Decorrelated* and *CELS* can be improved by using the procedures proposed in this paper.

This paper is organized as follows. In Section 2, the leaning process of a neural network is briefly analyzed. Moreover, *Decorrelated* and *CELS* are reviewed. In section 3, the proposed partitioning methodologies are described. The experimental setup is shown in section 4 whereas the results are in subsection 5.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 588–595, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Theoretical Background

2.1 Learning Process and Stopping Criteria

The network architecture employed in this paper is the *Multilayer Feedforward Network*, henceforth called *MF network*. In the experiments, the networks have been trained for a few iterations. In each iteration, the weights of the networks have been adapted with the *Backpropagation* algorithm by using all the patterns from the training set, T. At the end of the iteration the *Mean Square Error*, MSE, has been calculated by classifying all the patterns from the the Validation set, V. When the learning process has finished, the weights of the iteration with lowest MSE in the validation set are assigned to the final network.

2.2 Description of Ensemble Methodologies

Decorrelated: Two version of *Decorrelated*, *DECOv1* and *DECOv2*, were introduced by Rosen in [8]. In both versions, the networks are trained in serial and the main purpose is to penalize an individual classifier for being correlated with the previously trained one. For this reason Rosen added a *penalty term* (P) to the MSE equation (E) as in denoted in Eq. (1):

$$E^n(x) = \sum_{c=1}^{N_{cls}} \left(\frac{1}{2} \cdot \left(d_c(x) - y_c^n(x)\right)^2 + P_c^n(x) \right) \tag{1}$$

Where n stands for the number of network in the ensemble and c for the output class. The *Penalty* (Eq. 2) denotes the correlation degree between a network and the previously trained one.

$$P_c^n(x) = \lambda \cdot \left(d_c(x) - y_c^{n-1}(x)\right) \cdot \left(d_c(x) - y_c^n(x)\right) \tag{2}$$

Where λ denotes the weight of the penalty term which must be set empirically by trial and error because it depends on the classification problem. The networks are trained independently but the equations used in *Backpropagation* to update the weights of the *MF* networks have to be adapted to the new error equation.

Although both versions use the same penalty, *DECOv1* applies it to all the networks whereas *DECOv2* only introduces the penalty in the odd networks.

CELS: *Cooperative Ensemble Learning System* (*CELS*) is another ensemble variant that modifies the target equation and, therefore, the learning algorithm [6]. In this ensemble, all the networks of the ensemble are trained in parallel. Although the error is calculated with Eq.1, the penalty is given by:

$$P_c^n(x) = \lambda \cdot \left(y_c^n(x) - d_c(x)\right) \cdot \sum_{\substack{i=1 \\ i \neq n}}^{N_{nets}} \left(y_c^i(x) - d_c(x)\right) \tag{3}$$

Where λ is the weight of the penalty and it must be empirically set.

3 Creating New Specific Sets for Penalty Based Ensembles

3.1 Combining Penalties and Specific Sets

The ensembles *Decorrelated* and *CELS* modify the learning set by adding a penalty term to the target equation used for minimization. However, all the networks are trained using the same training and validation sets.

To perform the experiments, the original datasets, described in section 4.2, have been divided into three different subsets. The first set is the training set, T (64% of total patterns), which is used to adapt the weights of the networks. The second set is validation set, V (16% of total patterns), which is used to select the network configuration with the best estimated generalization capability. Finally, the last set is the test set, TS (20% of total patterns), which is used to get the accuracy of the network and the final results. When we refer to the original learning set, L, we really refer to the union of the training and validation sets.

In this paper, two different procedures (based on *Bagging* and *Cross-Validation*) are introduced to generate different training and validation sets for each network of these ensembles based on penalty terms. Diversity of the ensemble might be positively affected by using these procedures because the networks will not use exactly the same training set (used to adapt the weight values) and validation set (used to select the best network configuration with patterns not used for training).

Moreover, the new ensembles will take benefit from the use of the penalty terms introduced in *Decorrelated* and *CELS* because their aim is to reduce correlation of the networks.

3.2 Bagging as Set Generator

Although *Bootstrap Aggregating*, henceforth *Bagging*, is an ensemble model proposed by Breiman in [2], it can be used to generate specific training and validation sets. Concretely, the specific training set is generated for each network by randomly sampling patterns with replacement from the original learning set. According to reference [11], the generated training sets should double the original training set size (factor size $n = 2$). In this paper, this factor size will also be 1.5 times greater than the original training set size.

The patterns from the original learning set, L, which are not in the specific training set, T^{net} are chosen as patterns of the specific validation set, V^{net}.

The basic *Decorrelated* algorithm is modified according to Algorithm 1. The main difference between the original *Decorrelated* and the new *BagDecorrelated* is the inclusion of the first and second statements of the for loop. They were not in the original version and they have been included in *BagDecorrelated* to generate the specific training and validation sets according to *Bagging*.

All the networks are simultaneously trained in the original *CELS* algorithms. For this reason the original ensemble has been adapted to use specific training and validation sets. Concretely, two versions are introduced in Algorithms 2 and 3.

Algorithm 1. BagDecorrelated $\{T, V, N_{networks}\}$

```
for net = 1 to N_networks do
   Randomly Create T^net by sampling from L with replacement
   Generate V^net with the patterns from L which are not in T^net
   Set a random seed for wight initialization = seed_net
   MF Network Training {T^net , V^net}
end for
```

Algorithm 2. BagCELS-m1$\{T, V, N_{networks}\}$

```
for net = 1 to N_networks do
   Set initial weight values for net-network
   Randomly Create T^net by sampling from L with replacement
   Generate V^net with the patterns from L which are not in T^net
end for
for e = 1 to epochs do
   for i = 1 to n · N_learning do
      Select x as the i-esim element from learning set L
      for net = 1 to N_networks do
         Calculate output y^net(x)
      end for
      for net = 1 to N_networks do
         if x is in T^net then
            Adjust the trainable parameters
         end if
      end for
   end for
   for net = 1 to N_networks do
      Calculate MSE of network net with V^net
   end for
   Save ensemble configuration and MSE
end for
for net = 1 to N_networks do
   Select epoch with lowest validation MSE
   Assign the selected epoch configuration to net
   Save final network
end for
```

In the first version, *BagCELS-m1*, the networks are trained with the original learning set. For each epoch, all the patterns from L are presented to all the networks of the ensemble. Firstly, the output of an individual pattern, x, is calculated for all the networks of the ensemble. Then, the weights of a network, *net*, are adapted only if the pattern, x, is in the specific training set, T^{net}. In the other case, the weights are kept unchanged if the pattern is not in T^{net}. Finally, when all the patterns from the learning set have been presented to the networks, the MSE error is calculated for each network on the corresponding specific validation set, V^{net}.

Algorithm 3. BagCELS-m2$\{T, V, N_{networks}\}$

```
for net = 1 to N_networks do
    Random Generator Seed = seed_net
    Randomly Create T^net by sampling from L with replacement
    Generate V^net with the patterns from L which are not in T^net
end for
for e = 1 to epochs do
    for i = 1 to n · N_patterns do
        for net = 1 to N_networks do
            Select x as the i-esim element from learning set T^net
            for net2 = 1 to N_networks do
                Calculate output y^net2(x)
            end for
            Adjust the trainable parameters of net
        end for
        for net = 1 to N_networks do
            Calculate MSE of network net with V^net
        end for
    end for
    Save ensemble configuration and MSE
end for
for net = 1 to N_networks do
    Select epoch with lowest validation MSE
    Assign the selected epoch configuration to net
    Save final network
end for
```

In contrast, in the second adaptation (*BagCELS-m2*) each network is trained with its specific training set. For each pattern index, i, the $i-th$ pattern of the specific training set is used to adapt the weights for each network, *net*. To adapt these weights the outputs of the other networks on the pattern, i-esim element of T^{net}, have to be calculated. Finally, the MSE error is calculated for each network on the corresponding specific validation set as done in *BagCELS-m1*.

3.3 Cross Validation Committee as Set Generator

Similarly, the ensemble model *Cross-Validation Committee* can also be used to generate the specific training and validation sets. In this case, the original learning set is divided into N_{sets} subsets, L_i in eq. 4, and the specific training and validation sets are generated from them. The final network training set of network number *net* is T^{net} and its validation set is V^{net} according to the following equations.

$$T^{net} = \bigcup_{\substack{i=1 \\ i \neq index_{net,1} \\ i \neq index_{net,2}}}^{N_{sets}} L_i \qquad V^{net} = L_{index_{net,1}} \cup L_{index_{net,2}} \tag{4}$$

Where the indexes related to a neural network, $index_{net,1}$ and $index_{net,2}$, are randomly set with the constraint that the different networks have different training and validation sets. The value of N_{sets} has been set to 10 in order to keep the size of the original training and validation sets.

The base structure of *CVCv3Decorrelated*, *CVCv3CELSm1* and *CVCv3CELSm2* also corresponds to algorithms 1-3 but the procedure used to generate the specific sets is done according to *CVCv3* instead of *Bagging*.

4 Experimental Setup

4.1 Experiments

To test the performance of the proposed methods, ensembles of 3, 9, 20 and 40 networks have been built according to the original ensembles and the new *BagDecorrelated*, *BagCELS*, *CVCv3Decorrelated* and *CVCv3CELS*. The experiments have been repeated ten times with different partitions in the sets.

4.2 Description of the Databases

The following problems from the *UCI repository of machine learning databases* [1] have been used to test the performance of the methods: *Arrhythmia*, *Balance Scale*, *Cylinder Bands*, *BUPA liver disorders*, *Australian Credit Approval*, *Dermatology*, *Ecoli*, *Solar Flares*, *Glass Identification*, *Heart Disease*, *Image segmentation*, *Ionosphere Database*, *The Monk's Problem 1 and 2*, *Pima Indians Diabetes*, *Haberman's Survival Data*, *Congressional Voting Records*, *Vowel Database* and *Wisconsin Breast Cancer*. The optimal training parameters for these datasets have not been included due to the lack of space but they are publicly available in a Ph.D. thesis [10].

5 Results and Discussion

5.1 General Measurements

To perform an exhaustive comparison, the *mean Percentage of Error Reduction across all databases with respect to the Single Network* (*mean PER*) has been calculated to obtain the general behavior of the ensembles. A PER value of 0% means that there is no improvement in the percentage of correcly classfied patterns by the use of the ensemble method with respect to a single network whereas a positive value means that the ensembles is better than the single netowrk. Moreover, a negative value means that a single network performs better than the ensemble. The PER value is given by eq.5.

$$PER = 100 \cdot \frac{Perf_{Ensemble} - Perf_{Singlenet}}{100 - Perf_{SingleNet}} \tag{5}$$

Where $Perf_{singlenet}$ and $Perf_{ensemble}$ correspond to the percentage of patterns correctly classified by the single net and the ensemble respectively.

5.2 General Results

The main results for the new ensembles using *Bagging* to generate the specific sets are introduced in tables 1 and 2. Firstly, both versions of *DECO* and *CELS* have been improved by using *Bagging* as generator of the specific training and validation sets according to table 1.

Table 1. *Mean PER - Bagging* as set generator

ensemble	*n*	*3-Net*	*9-Net*	*20-Net*	*40-Net*
DECOv1	-	24.73	26.63	26.84	27.09
BagDECOv1	1.5	22.22	28.33	28.54	28.92
BagDECOv1	2	24.80	26.92	29.62	29.19
DECOv2	-	24.91	25.73	25.93	26.40
BagDECOv2	1.5	22.26	26.99	28.18	28.96
BagDECOv2	2	25.71	28.79	29.25	29.63
CELS	-	21.51	23.73	25.75	26.35
BagCELS-m1	1.5	21.84	27.16	26.71	24.39
BagCELS-m1	2	21.71	27.24	29.00	28.20
BagCELS-m2	1.5	21.76	25.66	26.70	24.75
BagCELS-m2	2	24.61	28.15	30.16	28.25

Secondly, the best results of *BagDecorrelated* are obtained when n is set to value 2. For *BagCELS-m1* the best value of n depends on the ensemble size , n equal to 1.5 for 3 networks and n equal to 2 for the other three cases. The best values for the factor n in *BagCELS-m2* is 2.

Thirdly, the best results are provided by *BagDECOv2* for ensembles of 3, 9 and 40 networks. For 20 networks, the best approach is *BagCELS-m2*. Furthermore, the "worst" traditional ensemble is *CELS* but *CVCv3CELS-m2* provides the best overall results. Generating specific sets should be seriously considered.

Table 2. *Mean PER - Cross-Validated Committee v3* as set generator

ensemble	*3-Net*	*9-Net*	*20-Net*	*40-Net*
CVCv3DECOv1	24.64	29.07	28.84	29.20
CVCv3DECOv2	24.42	28.25	29.79	29.77
CVCv3CELS-m1	25.31	27.32	28.23	27.52
CVCv3CELS-m2	23.71	26.32	27.33	27.70

According to table 2, the new ensembles based on *CVCv3* improve in general the results of the original ensembles. *CVCv3CELS-m1* is the best ensemble for 3 nets whereas *CVCv3DECOv1* (for 9 nets) and *CVCv3DECOv2* (for 20 and 40 nets) are a better choice.

Secondly, *CVCv3DECOv1* is better than *CVCv3DECOv2* for 3 and 9 networks but *CVCv3DECOv2* with 20 networks provides the best overall results.

Finally, *CVCv3CELS-m1* and *CVCv3CELS-m2* are better than the original *CELS* for all the cases. Moreover, *CVCv3CELS-m1* is more suitable for ensembles of 3 to 20 networks whereas *CVCv3CELS-m2* fits better for 40 networks.

6 Conclusions

Some traditional ensembles (*DECOv1*, *DECOv2* and *CELS*) have been successfully fused with *Bagging* and *Cross-Validation Committee*. In general, the original ensembles have been improved by using specific training and validation sets to train each network of the ensemble. In fact, the worst traditional ensemble was *CELS* but the best overall results are provided by *BagCELS-m2*. Moreover, the new methods outperform the best results of the traditional ensembles with less networks which can be useful when computational resources are critical.

Between the two alternatives to generate the specific sets, *Bagging* provides better results than *CVCv3* in 62.5% of the total cases and provides the best overall results (20 networks and *BagCELS-m2*). However, there are specific cases in which *CVCv3* is more suitable. For this reason, both procedures should be seriously considered to use with traditional ensemble methods based on penalties.

References

1. Asuncion, A., Newman, D.: UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences (2007)
2. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
3. Fernández-Redondo, M., Hernández-Espinosa, C., Torres-Sospedra, J.: Multilayer feedforward ensembles for classification problems. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) ICONIP 2004. LNCS, vol. 3316, pp. 744–749. Springer, Heidelberg (2004)
4. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning, pp. 148–156 (1996)
5. Hernández-Espinosa, C., Torres-Sospedra, J., Fernández-Redondo, M.: New experiments on ensembles of multilayer feedforward for classification problems. In: Proceedings of IJCNN 2005, pp. 1120–1124 (2005)
6. Liu, Y., Yao, X.: Simultaneous training of negatively correlated neural networks in an ensemble. IEEE T. Syst. Man. Cyb. 29, 716 (1999)
7. Parmanto, B., Munro, P.W., Doyle, H.R.: Improving committee diagnosis with resampling techniques. In: Advances in Neural Information Processing Systems, pp. 882–888 (1996)
8. Rosen, B.E.: Ensemble learning using decorrelated neural networks. Connection Science 8(3-4), 373–384 (1996)
9. Torres-Sospedra, J., Hernández-Espinosa, C., Fernández-Redondo, M.: Adaptive boosting: Dividing the learning set to increase the diversity and performance of the ensemble. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) ICONIP 2006. LNCS, vol. 4232, pp. 688–697. Springer, Heidelberg (2006)
10. Torres-Sospedra, J.: Ensembles of Artificial Neural Networks: Analysis and Development of Design Methods. Ph.D Thesis, Universitat Jaume I (2011)
11. Tumer, K., Ghosh, J.: Error correlation and error reduction in ensemble classifiers. Connection Science 8(3-4), 385–403 (1996)
12. Yildiz, O.T., Alpaydin, E.: Ordering and finding the best of k>2 supervised learning algorithms. IEEE T. Pattern. Anal. 28(3), 392–402 (2006)

A New Algorithm for Learning Mahalanobis Discriminant Functions by a Neural Network

Yoshifusa Ito[1], Hiroyuki Izumi[2], and Cidambi Srinivasan[3]

[1] School of Medicine, Aichi Medical University
Nagakute, Aichi-ken, 480-1195 Japan
ito@aichi-med-u.ac.jp
[2] Department of Policy Science, Aichi-Gakuin University
Nisshin, Aichi-ken, 470-0195 Japan
hizumi@psis.aichi-gakuin.ac.jp
[3] Department of Statistics, University of Kentucky
Patterson Office Tower, Lexington, Kentucky 40506, USA
srini@ms.uky.edu

Abstract. It is well known that a neural network can learn Bayesian discriminant functions. In the two-category normal-distribution case, a shift by a constant of the logit transform of the network output approximates a corresponding Mahalanobis discriminant function [7]. In [10], we have proposed an algorithm for estimating the constant, but it requires the network to be trained twice, in one of which the teacher signals must be shifted by the mean vectors. In this paper, we propose a more efficient algorithm for estimating the constant with which the network is trained only once.

1 Introduction

The Mahalanobis and Bayesian discriminant functions are based on the distinct concepts; the former on the distances and the latter on the probabilities. Nevertheless, they are closely related in the two-category normal-distribution case [7], [10]. In this paper we use this relation.

The Mahalanobis distance is commonly used in the discriminant analysis. However, there is no well-known efficient algorithm for neural networks to approximate this discriminant function. On the other hand, it is well known that a neural network can learn Bayesian discriminant functions [2], [11-13]. In [2], Funahashi proposed a neural network to approximate the Bayesian discriminant function in the two-category normal-distribution case. The activation function of the output unit of his network is the logistic function and, hence, its inner potential approximates a quadratic form. In [7], we have remarked that if the inner potential is shifted by a constant, the network output approximates a corresponding Mahalanobis discriminant function.

Later we proposed an algorithm for estimating the constant with a neural network, implying that the network can learn a Mahalanobis discriminant function [10]. However, the network must learn two types of Bayesian discriminant functions to this end. The first is to estimate the constant. In this paper, we propose

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 596–605, 2011.
© Springer-Verlag Berlin Heidelberg 2011

a simpler but more efficient algorithm. When this is applied, it is unnecessary for the network to be trained twice. Instead, the network must be equipped with three additional memory nodes.

In simulations, we use neural networks based on [3]. Accordingly, the network has more hidden units than the minimum requirement but has less inner parameters of the hidden-layer activation functions. This makes learning easier.

2 Mahalanobis and Bayesian Discriminant Functions

2.1 Preliminaries

In this paper we treat discriminant analysis in two-category, normal-distribution cases. We denote by θ_i, $i = 1, 2$, the categories and by $N(\mu_i, \Sigma_i)$ the normal distributions, where μ_i are the mean vectors and Σ_i the covariance matrices. Furthermore, we denote by $p(x|\theta_i)$ the state-conditional probability distributions:

$$p(x|\theta_i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} e^{-\frac{1}{2}(x-\mu_i)^t \Sigma_i^{-1}(x-\mu_i)}, i = 1, 2, \quad x \in \mathbf{R}^d. \tag{1}$$

The Mahalanobis distance $d(x, \mu_i)$ from x to μ_i is defined by

$$d_i(x, \mu_i) = |(x-\mu_i)^t \Sigma_i^{-1}(x-\mu_i)|^{1/2}, \quad i = 1, 2.$$

Hence, if

$$|(x-\mu_1)^t \Sigma_1^{-1}(x-\mu_1)|^{1/2} < |(x-\mu_2)^t \Sigma_2^{-1}(x-\mu_2)|^{1/2}, \tag{2}$$

x is allocated to the category θ_1 and vice versa. This classification is equivalent to classifying x by a discriminant function

$$q_M(x) = -\frac{1}{2}\{(x-\mu_1)^t \Sigma_1^{-1}(x-\mu_1) - (x-\mu_2)^t \Sigma_2^{-1}(x-\mu_2)\}. \tag{3}$$

If $q_M(x) > 0$, then x is allocated to θ_1. Let σ be the logistic function $\sigma(t) = \frac{1}{1+e^{-t}}$. Since this function is monotone,

$$\psi_M(x) = \sigma(q_M(x)) \tag{4}$$

is also a Mahalanobis discriminant function. If $\psi_M(x) > 0.5$, x is allocated to the category θ_1.

Let $P(\theta_i|x)$, $i = 1, 2$, be the posterior probabilities of the respective categories. Then, each of them can be used as a Bayesian discriminant function. It is well known that a neural network can approximate the posterior probabilities [2, 3-13]. For convenience we set

$$\psi_B(x) = P(\theta_1|x). \tag{5}$$

If $\psi_B(x) > 0.5$, x is allocated to the category θ_1. Furthermore, any monotone transforms of $P(\theta_i|x)$ can be used as Bayesian discriminant functions [1]. Among them we are in particular concerned with its logit transform:

$$q_B(x) = \sigma^{-1}(\psi_B(x)) = \log \frac{P(\theta_1|x)}{P(\theta_2|x)} = \log \frac{P(\theta_1)}{P(\theta_2)} + \log \frac{p(x|\theta_1)}{p(x|\theta_2)} \tag{6}$$

where $P(\theta_i)$, $i = 1, 2$, are the prior probabilities of the categories. We denote by $\hat{\psi}_B$ the approximation of ψ_B estimated by a neural network and set $\hat{q}_B = \sigma^{-1}(\hat{\psi}_B)$. The inner potential of Funahashi's network realizes this function [2].

2.2 Conversion of a Bayesian Discriminant Function to a Mahalanobis Discriminant Function

In the two-category, normal-distribution case, q_B is a quadratic form:

$$q_B(x) = \log \frac{P(\theta_1)}{P(\theta_2)} - \frac{1}{2} \log \frac{|\Sigma_1|}{|\Sigma_2|} - \frac{1}{2}\{(x-\mu_1)^t \Sigma_1^{-1}(x-\mu_1) - (x-\mu_2)^t \Sigma_2^{-1}(x-\mu_2)\}. \tag{7}$$

In [7], we remarked that

$$q_B(x) = \log \frac{P(\theta_1)}{P(\theta_2)} - \frac{1}{2} \log \frac{|\Sigma_1|}{|\Sigma_2|} + q_M(x) \tag{8}$$

and the difference between q_B and q_M is a constant

$$C = \log \frac{P(\theta_1)}{P(\theta_2)} - \frac{1}{2} \log \frac{|\Sigma_1|}{|\Sigma_2|}. \tag{9}$$

This constant can be used to convert the Bayesian discriminant function q_B to the Mahalanobis discriminant function q_M: $q_M(x) = q_B(x) - C$. Since σ is monotone, $\psi_M = \sigma(q_M)$ is also a Mahalanobis discriminant function. We have $\sigma(q_M(x)) = \sigma(q_B(x) - C)$; that is,

$$\psi_M(x) = \sigma(\sigma^{-1}(\psi_B(x) - C). \tag{10}$$

Our proposed simple algorithm for estimating C with a neural network is based on the following reasoning. By (7) and (9), we have

$$q_B(\mu_1) = C + \frac{1}{2}(\mu_1 - \mu_2)^t \Sigma_2^{-1}(\mu_1 - \mu_2), \tag{11}$$

$$q_B(\mu_2) = C - \frac{1}{2}(\mu_1 - \mu_2)^t \Sigma_1^{-1}(\mu_1 - \mu_2) \tag{12}$$

and

$$q_B(\frac{\mu_1 + \mu_2}{2}) = C - \frac{1}{8}(\mu_1 - \mu_2)^t (\Sigma_1^{-1} - \Sigma_2^{-1})(\mu_1 - \mu_2). \tag{13}$$

From these we obtain

$$C = 2q_B(\frac{\mu_1 + \mu_2}{2}) - \frac{1}{2}(q_B(\mu_1) + q_B(\mu_2)). \tag{14}$$

By (6), the logit transform of ψ_B is q_B. Hence, if a network is trained so that it outputs an approximation $\hat{\psi}_B$ of a Bayesian discriminant function ψ_B, an approximation $\hat{C}$ of C can be obtained by

$$\hat{C} = 2\hat{q}_B(\frac{\mu_1 + \mu_2}{2}) - \frac{1}{2}(\hat{q}_B(\mu_1) + \hat{q}_B(\mu_2)). \tag{15}$$

3 Construction of the Neural Network

We use the idea in [3] to construct the neural network. There are unit vectors v_i, $i = 1, ..., \frac{1}{2}d(d+1)$, in $\mathbf{R}^d$ for which the squares of the inner products $(v_i \cdot x)^2$ are linearly independent. Any homogeneous quadratic form in $\mathbf{R}^d$ can be expressed as a linear sum of the squares. These unit vectors are used in our neural network, but fixed beforehand. Let g be a twice continuously differentiable nonzero function defined on $\mathbf{R}$ such that $g^{(2)}(0) \neq 0$. Then, for the probability measure p, which is defined by

$$p(x) = P(\theta_1)p(x|\theta_1) + P(\theta_2)p(x|\theta_2),$$

and any quadratic form q in $\mathbf{R}^d$, there exist constants a_i, $i = 1, ..., \frac{1}{2}d(d+1)$, b_i, $i = 1, ..., d$, c, δ for which

$$\bar{q}(x) = \sum_{i=1}^{\frac{1}{2}d(d+1)} a_i g(\delta v_i \cdot x) + \sum_{i=1}^{d} b_i \cdot x_i + c \tag{16}$$

approximates q with any accuracy in the sense of $L^2(\mathbf{R}^d, p)$ [3].

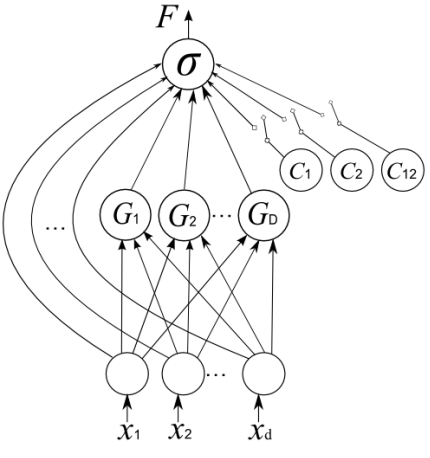


Fig. 1. A one-hidden-layer neural network with direct connections between the input layer and the output unit. This network has additional nodes C_1, C_2 and C_{12}. Here, $D = \frac{1}{2}d(d+1)$.

The formula (16) can be realized by our neural network having the structure illustrated in Fig.1, where $D = \frac{1}{2}d(d+1)$. The nodes marked by C_1, C_2 and C_{12} are memory nodes to memorize the estimated values of the constants defined by (11), (12) and (13) and are used for conversion of the approximation of the Bayesian discriminant function to that of the Mahalanobis discriminant function. For realization of (16), these nodes are unnecessary. The activation function of the nodes marked by G_i, $i = 1, ..., D = \frac{1}{2}d(d+1)$ is the function g and that of the output unit is σ. This network has direct connections between the input layer and the output unit. They are to approximate the linear part in (16).

4 Training of the Neural Network

Let $F(x, w)$ be the output of the network, where w is the weight vector. For an integrable function $\xi(x, \theta)$ defined on $\mathbf{R}^d \times \{\theta_1, \theta_2\}$, let $E[\xi(x, \cdot)|x]$ and $V[\xi(x, \cdot)|x]$ be its conditional expectation and variance. The proposition below is proved in [12] and has been used by many authors [2], [4-11], [13].

Set

$$\mathcal{E}(w) = \int_{\mathbf{R}^d} \sum_{i=1}^{2} (F(x,w) - \xi(x,\theta_i))^2 P(\theta_i) p(x|\theta_i) dx. \tag{17}$$

Then,

$$\mathcal{E}(w) = \int_{\mathbf{R}^d} (F(x,w) - E[\xi(x,\cdot)|x])^2 p(x) dx + \int_{\mathbf{R}^d} V[\xi(x,\cdot)|x] p(x) dx. \tag{18}$$

If $\xi(x,\theta_1) = 1$ and $\xi(x,\theta_2) = 0$, then $E[\xi(x,\cdot)|x] = P(\theta_1|x)$. Hence, when $\mathcal{E}(w)$ is minimized, the output $F(x,w)$ is expected to approximate $\psi_B(x) = P(\theta_1|x)$. Then, the inner potential of the output unit is expected to approximate ψ_B. The network is trained by minimizing

$$\mathcal{E}_n(w) = \frac{1}{n} \sum_{k=1}^{n} (F(x^{(k)},w) - \xi(x^{(k)},\theta^{(k)}))^2, \tag{19}$$

where $\{(x^{(k)},\theta^{(k)})\}_{k=1}^n \subset \mathbf{R}^d \times \{\theta_1,\theta_2\}$ is the teacher sequence.

If the means μ_1, μ_2 and their average $\frac{\mu_1+\mu_2}{2}$ are fed in turn to the network, then the inner potential of the network of Funahashi's type approximates (11), (12) and (13). These are respectively memorized in the nodes marked by C_1, C_2 and C_{12} in Fig.1. Hence, by (14), the inner potential approximates the Mahalanobis discriminant function $q_M(x)$, if these nodes are connected to the output unit with weights $-\frac{1}{2}$, $-\frac{1}{2}$ and 2. In the case where the network is not of Funahashi's type, the logit transform of the output can be used in the same way.

5 Simulations

Throughout this section, $f_1(x) = p(x|\theta_1)$, $f_2(x) = p(x|\theta_2)$ and ψ_B, ψ_M are as defined before. We denote by $\hat{\psi}_B$ and $\hat{\psi}_M$ the Bayesian and Mahalanobis discriminant functions obtained by simulation. Here, $\hat{\psi}_B$ is the network output after training and $\hat{\psi}_M$ is obtained by

$$\hat{\psi}_M(x) = \sigma(\sigma^{-1}(\hat{\psi}_B(x) - \hat{C}), \tag{20}$$

where $\hat{C}$ is defined by (15). The meaning of this equation may be obvious when compared with (10). In applications, the estimated means $\hat{\mu}_1$, $\hat{\mu}_2$ must be used, but the purpose of the simulation is to show that the algorithm works well. Accordingly, we use the real means μ_1, μ_2. We have repeated simulations many times changing the parameters, the number of teacher signals and the seed for the random numbers. We show here part of the results.

5.1 One-Dimensional Case

The parameters we have used are listed in **Table 1**. The patterns of the probability distributions, the theoretically calculated Bayesian and Mahalanobis discriminant functions ψ_B, ψ_M and their difference in each example based on the parameters are illustrated in **Fig. 2**.

Table 1. The parameters used in the one-dimensional examples

	$P(\theta_1)$	$P(\theta_2)$	μ_1	μ_2	σ_1^2	σ_2^2
Example 1	0.6	0.4	2	-2	1	2
Example 2	0.7	0.3	2	-1	1	3

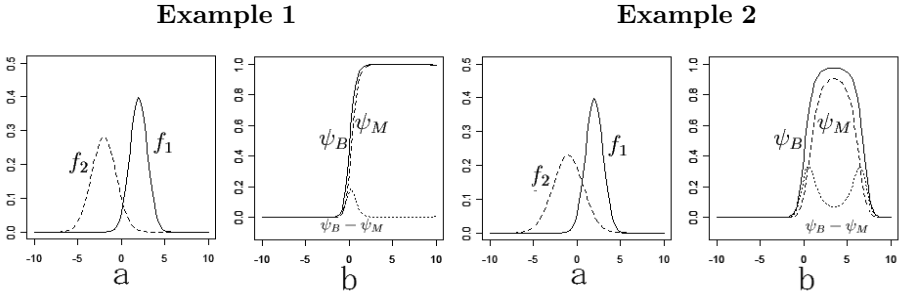


Fig. 2. Patterns of the functions based on the parameters in **Table 1. a**: The state-conditional probability distributions f_1 and f_2. **b**: The Bayesian and Mahalanobis discriminant functions ψ_B and ψ_M with their difference $\psi_B - \psi_M$.

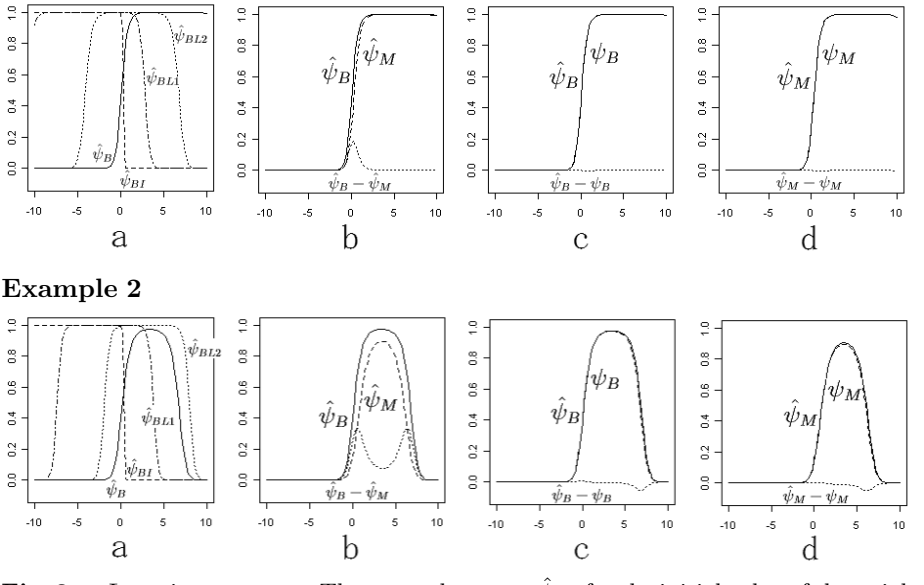


Fig. 3. a: Learning processes. The network output $\hat{\psi}_{BI}$ for the initial value of the weight vector converges to $\hat{\psi}_B$ via $\hat{\psi}_{BL1}$ and $\hat{\psi}_{BL2}$ while training. **b**: $\hat{\psi}_B$ is compared with $\hat{\psi}_M$ with their difference. **c**: $\hat{\psi}_B$ is compared with ψ_B. **d**: $\hat{\psi}_M$ is compared with ψ_M.

In each example, the network was trained with a sequence of independent 1000 teacher signals having the parameters in **table 1**. The learning processes are illustrated in **Fig.3**, where $\hat{\psi}_{BI}$ is the network output for the initial value of the weight vector. Via $\hat{\psi}_{BL1}$ and then $\hat{\psi}_{BL2}$, it converged to $\hat{\psi}_B$. This is the Bayesian discriminant function obtained by simulation. From this, $\hat{\psi}_M$ is obtained by (20) as stated above. In Example 1, $\hat{C} = 0.752$ and $C = 0.752$ and, in Example 2, $\hat{C} = 1.42$ and $C = 1.40$. In this figure, the two functions $\hat{\psi}_B$ and $\hat{\psi}_M$ are mutually compared, and they are respectively compared with their theoretical counterparts ψ_B and ψ_M.

Table 2. Allocation results of 1000 test signals by the discriminant functions. **Sgls** implies test signals, and **Correct** the numbers of signals correctly allocated.

Example 1

Category	Sgls	ψ_B	ψ_M	$\hat{\psi}_B$	$\hat{\psi}_M$
θ_1	619	624	617	624	617
θ_2	381	376	383	376	383
Correct		939	939	939	934

Example 2

Category	Sgls	ψ_B	ψ_M	$\hat{\psi}_B$	$\hat{\psi}_M$
θ_1	697	764	655	764	653
θ_2	303	236	345	236	347
Correct		891	891	891	866

The four discriminant functions ψ_B, ψ_M, $\hat{\psi}_B$, $\hat{\psi}_M$ are respectively tested with mutually independent 1000 test signals generated from the sources with the parameters in **Table 1.** They are independent from the teacher signals. The allocation results by the four are shown in **Table 2**. In Example 1, each discriminant function allocated more than 93 % of signals correctly but, in Example 2, the allocation accuracy is a little worse. However, the allocations by $\hat{\psi}_B$ and ψ_B, and by ψ_M and $\hat{\psi}_M$ in Example 1 coincided at all 1000 test signals, and those by $\hat{\psi}_B$ and ψ_B in Example 2 also coincided at 1000 signals, but by $\hat{\psi}_M$ and ψ_M at 998 test signals. These can be expected from **Fig.3** where the discrepancies between the theoretical discriminant functions and the corresponding network outputs are small.

5.2 Two-Dimensional Case

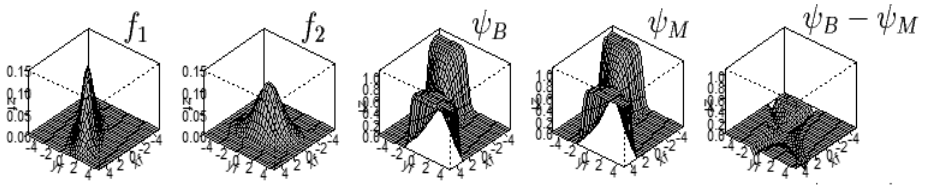
In the two examples in the two-dimensional case, the parameters in **Table 3** were used. The probability distributions and the discriminant functions ψ_B and ψ_M based on these parameters are illustrated **Fig.4** with the differences $\psi_B - \psi_M$ as in the one-dimensional case.

Table 3. The parameters used in the two-dimensional examples

	$P(\theta_1)$	$P(\theta_2)$	μ_1	μ_2	Σ_1	Σ_2
Example 1	0.3	0.7	(1, 0)	(0, 0)	$\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & -0.3 \\ -0.3 & 2 \end{pmatrix}$
Example 2	0.5	0.5	(0, -0.8)	(0.1, 0)	$\begin{pmatrix} 2 & 0.8 \\ 0.8 & 2 \end{pmatrix}$	$\begin{pmatrix} 1.1 & -0.1 \\ -0.1 & 1.1 \end{pmatrix}$

Distinct from the one-dimensinal case, 1000 teacher signals are insufficient. Hence, we illustrate here the results respectively obtained with 5000 teacher signals. The learning processes are shown in **Fig.5** with $\hat{\psi}_M$, calculated by (20), and the differences $\hat{\psi}_B - \hat{\psi}_M$. The initial network outputs $\hat{\psi}_{BI}$ converge to $\hat{\psi}_B$ via $\hat{\psi}_{BL}$ respectively. In Example 1, $\hat{C} = -0.503$ and $C = -0.524$ and, in Example 2, $\hat{C} = -0.514$ and $C = -0.515$. The discriminant functions $\hat{\psi}_B$ and $\hat{\psi}_M$ are compared respectively with their counterpart in **Fig.6.**

Example 1



Example 2

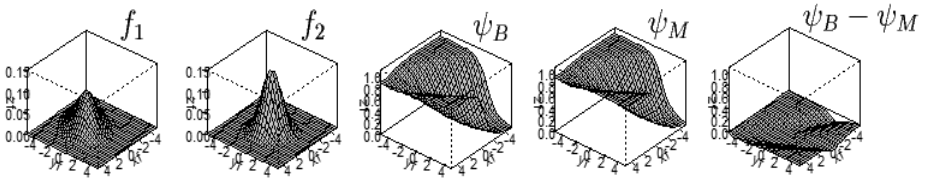
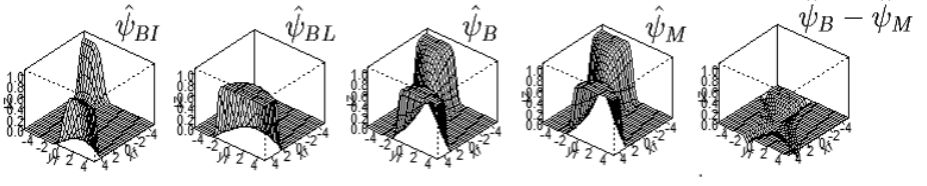


Fig. 4. The probability distributions and the discriminant functions with the parameters in **Table 3**

Example 1



Example 2

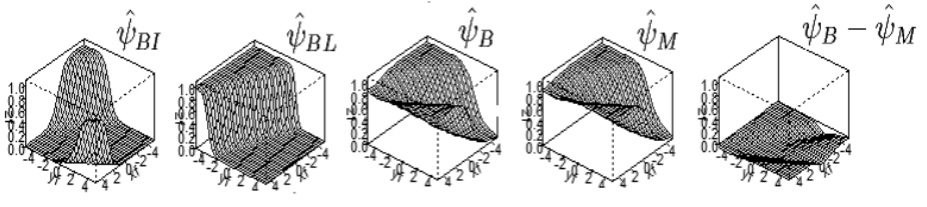


Fig. 5. Learning process is shown by $\hat{\psi}_{BI}$, $\hat{\psi}_{BL}$ and $\hat{\psi}_B$. For comparison, $\hat{\psi}_M$ and $\hat{\psi}_B - \hat{\psi}_M$ are also shown.

The discriminant functions obtained by simulation are tested with 1000 test signals as in the one-dimensional case. The results are summarized in **Table 4**.

In Example 1, the allocation results of the respective discriminant functions were correct at about 78% of the test signals, and, in Example 2, the results were a little worse. However, in Example 1, the allocations results by $\hat{\psi}_B$ and ψ_B coincided at more than 99% of signals and those by $\hat{\psi}_M$ and ψ_M also at more than 99%. In Example 2, those by $\hat{\psi}_B$ and ψ_B coincided at 989 signals and by $\hat{\psi}_M$ and ψ_M at more than 99% signals.

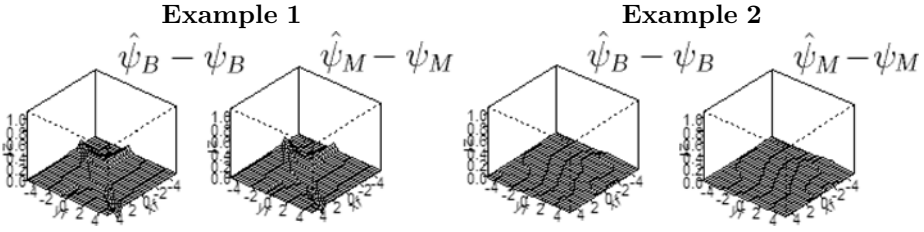


Fig. 6. Differences of the discriminant functions obtained by simulation and the respective corresponding theoretical discriminant functions

Table 4. Allocation results by the four discriminant functions ψ_B, ψ_M, $\hat{\psi}_B$ and $\hat{\psi}_M$

Example 1

Category	Sgls	ψ_B	ψ_M	$\hat{\psi}_B$	$\hat{\psi}_M$
θ_1	288	194	288	195	301
θ_2	477	473	535	483	541
Correct		786	786	787	755

Example 2

Category	Sgls	ψ_B	ψ_M	$\hat{\psi}_B$	$\hat{\psi}_M$
θ_1	483	416	677	409	679
θ_2	517	584	323	591	321
Correct		669	669	668	630

6 Discussions

In applications, the constant $\hat{C}$ must be calculated with (15) where μ_1 and μ_2 are replaced by the estimated means $\hat{\mu}_1$ and $\hat{\mu}_2$ obtained from the teacher sequence. We have used these means, too, but the allocation results did not much changed. This is expected because the constants $\hat{C}$ estimated by this way are almost the same as those estimated by the present method. To save space we have to omit the details.

In the one-dimensional case, the allocation results by the discriminant functions $\hat{\psi}_B$ and $\hat{\psi}_M$, obtained with 1000 teacher signals, coincided with those by ψ_B and ψ_M respectively at almost all test signals. In the two-dimensional case, the score was not so good with the same number of the teacher signals. However, when the number was increased to 5000, it was remarkably improved as described above, implying that the algorithm worked.

There is another method of approximating the Mahalanobis discriminant functions [10]. However, it is more complicated. In [10], the network must be trained twice. If the mean vectors of teacher signals are zero, the network approximates

$$q_{B0}(x) = \log \frac{P(\theta_1)}{P(\theta_2)} - \frac{1}{2} \log \frac{|\Sigma_1|}{|\Sigma_2|} - \frac{1}{2} x^t (\Sigma_1^{-1} - \Sigma_2^{-1}) x.$$

From this equation, we have $C = q_{B0}(0) = \log \frac{P(\theta_1)}{P(\theta_2)} - \frac{1}{2} \log \frac{|\Sigma_1|}{|\Sigma_2|}$. Hence, if the network is trained with teacher signals $\{(x - \hat{\mu}_i, \theta_i)\}$, $i = 1, 2$, it approximates ψ_{B0}. Hence, the approximation of the constant can be obtained by $\hat{C} = \hat{q}_{B0}(0) = \sigma^{-1}(\hat{\psi}_{B0}(0))$. The right-hand side is the inner potential of the output unit for $x = 0$.

This method is complicated in the sense that the mean vectors must be estimated beforehand and, then, the network is trained with the teacher sequence $\{(x - \hat{\mu}_i, \theta_i)\}$ and, further, the network is trained with the original teacher signals $\{(x, \theta_i)\}$; that is, the network must be trained twice. The present method is more efficient, because the network is to be trained only once.

Acknowledgement. This work was supported by a Grant-in-Aid for Scientific Research (22500213) from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. Duda, R.O., Hart, P.E.: Pattern classification and scene analysis. John Wiley & Sons, New York (1973)
2. Funahashi, K.: Multilayer neural networks and Bayes decision theory. Neural Networks 11, 209–213 (1998)
3. Ito, Y.: Simultaneous approximations of polynomials and derivatives and their applications to neural networks. Neural Computation 20, 2757–2791 (2008)
4. Ito, Y., Srinivasan, C.: Multicategory Bayesian Decision Using a Three-Layer Neural Network. In: Kaynak, O., Alpaydın, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714, pp. 253–261. Springer, Heidelberg (2003)
5. Ito, Y., Srinivasan, C.: Bayesian decision theory on three-layer neural networks. Neurocomputing 63, 209–228 (2005)
6. Ito, Y., Srinivasan, C., Izumi, H.: Bayesian Learning of Neural Networks Adapted to changes of Prior Probabilities. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3697, pp. 253–259. Springer, Heidelberg (2005)
7. Ito, Y., Srinivasan, C., Izumi, H.: Discriminant Analysis by a Neural Network with Mahalanobis Distance. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4132, pp. 350–360. Springer, Heidelberg (2006)
8. Ito, Y., Srinivasan, C., Izumi, H.: Learning of Bayesian Discriminant Functions by a Neural Network. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part I. LNCS, vol. 4984, pp. 238–247. Springer, Heidelberg (2008)
9. Ito, Y., Srinivasan, C., Izumi, H.: Multi-Category Bayesian Decision by Neural Networks. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) ICANN 2008, Part I. LNCS, vol. 5163, pp. 21–30. Springer, Heidelberg (2008)
10. Ito, Y., Izumi, H., Srinivasan, C.: Learning of Mahalanobis Discriminant Functions by a Neural Network. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009. LNCS, vol. 5863, pp. 417–424. Springer, Heidelberg (2009)
11. Richard, M.D., Lipmann, R.P.: Neural network classifiers estimate Bayesian a posteriori probabilities. Neural Computation 3, 461–483 (1991)
12. Ruck, M.D., Rogers, S., Kabrisky, M., Oxley, H., Sutter, B.: The multilayer perceptron as approximator to a Bayes optimal discriminant function. IEEE Transactions on Neural Networks 1, 296–298 (1990)
13. White, H.: Learning in artificial neural networks: A statistical perspective. Neural Computation 1, 425–464 (1989)

Learning of Dynamic BNN toward Storing-and-Stabilizing Periodic Patterns

Ryo Ito, Yuta Nakayama, and Toshimichi Saito*

Hosei University, Koganei, Tokyo, 184-8584 Japan
http://www.hosei.ac.jp

Abstract. This paper studies learning algorithm of a dynamic binary neural network having rich dynamics. The algorithm is based on the genetic algorithm with an effective kernel chromosome and hidden neuron sharing. Performing basic numerical experiments, we have confirmed that the algorithm can store desired periodic teacher signals and the stored signals are stable for initial value.

Keywords: Binary neural networks, genetic algorithms, learning.

1 Introduction

Binary neural networks (BNN) are three-layer feed-forward artificial neural networks that transforms N-dimensional (N-D) binary input to M-D binary output. The BNN has the signum activation function that is suitable to treat Boolean functions in the case $M = 1$ whereas the MLP is suitable to treat smooth functions. Several learning algorithms have been studied for storing desired binary teacher signals: the Boolean-like training [1], the expand-and-truncate learning [2] [4], the DNA-like learning [5], etc. The applications are many, including synthesis of logical circuits [3], cellular automata [6] [7], PWM signals [8]-[10], etc.

This paper studies dynamic binary neural networks (DBNN) and its learning algorithm (GALA) based on the genetic algorithms (GA, [7] [11]). Basically, the DBNN is constructed by applying the feedback with delay to the BNN with $M = N$. The parameters are simplified: weighting in hidden neuron is ternary and that in output neuron is 1 or 0. The DBNN can generate various binary vector sequence (BVS), hence it is suitable to treat dynamic teacher signals. The GALA has two features: one of teacher signals is used as an initial chromosome (kernel) and a hidden neuron can be shared by plural output neurons. These are effective to reduce the number of hidden neurons and to simplify the network structure. Performing basic numerical experiments for storing periodic BVS, we have confirmed that the GALA can store the desired BVSs in the DBNN and that the stored BVS is stable for initial value: the GALA is applied to store the BVS and the BVS can be stable automatically. Although this stabilization function is confirmed only in two examples, it can be a trigger to develop novel learning algorithms with stabilization function. Note that Ref. [10] presents a basic version of GALA, however, it does not use the initial kernel and produces redundant hidden neurons. Also, it does not discuss the stabilization functions.

* This work is supported in part by JSPS KAKENHI#21500223.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 606–611, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Dynamic Binary Neural Networks

First, we define the BNN with simplified parameters. It transforms an N-D binary input $\boldsymbol{x} \equiv (x_1, \cdots, x_N)$ to an M-D binary output $\boldsymbol{y} \equiv (y_1, \cdots, y_M)$:

$$\begin{aligned} y_k &= \mathrm{sgn}\left(\textstyle\sum_{j=1}^{N_H} w_{kj}^o \xi_j - T_k^o\right), \; w_{kj}^o \in \{0,1\}, \qquad T_k^o = \textstyle\sum_{j=1}^{N_H} 1 - |w_{kj}^o| \\ \xi_j &= \mathrm{sgn}\left(\textstyle\sum_{i=1}^{N} w_{ji} x_i - T_j\right), \quad w_{ji} \in \{-1,0,1\}, T_j = \textstyle\sum_{i=1}^{N} |w_{ji}| - \beta_j \end{aligned} \tag{1}$$

where $x_i \in \boldsymbol{B} \equiv \{-1, 1\}$, $i = 1 \sim N$, $y_k \in \boldsymbol{B}$, $k = 1 \sim M$ and β_j is a positive odd integer. $\boldsymbol{\xi} \equiv (\xi_1, \cdots, \xi_{N_H})$ is an N_H-dimensional hidden output where $\xi_j \in \boldsymbol{B}$, $j = 1 \sim N_H$. The signum activation function is defined by $\mathrm{sgn}(x) = 1$ for $x \geq 0$ and $\mathrm{sgn}(x) = -1$ for $x < 0$. The number of hidden neurons N_H can be used to measure the network simplicity. Eq. (1) is abbreviated by $\boldsymbol{y} = F_b(\boldsymbol{x})$.

We overview the case of single output ($M = 1$). If $\beta_j = 1$ then the j-th hidden neuron is equivalent to conjunction of x_i with nonzero weighting ($w_{ji} \neq 0$). The output neuron is equivalent to disjunction of ξ_j with connection $w_{1j}^o = 1$. In this case F_B is equivalent to the disjunctive canonical form (DCF) of Boolean functions. For example, the DCF $y = \bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_3 + \bar{x}_1\bar{x}_2$ is equivalent to the BNN with $N_H = 3$ in Fig. 1 (a): $y = \mathrm{sgn}(\xi_1 + \xi_2 + \xi_3 + 2)$, $\xi_1 = \mathrm{sgn}(-x_2 - x_3 - 1)$, $\xi_2 = \mathrm{sgn}(-x_1 - x_3 - 1)$ and $\xi_3 = \mathrm{sgn}(-x_1 - x_2 - 1)$ where $\beta_1 = \beta_2 = \beta_3 = 1$. If $\beta_j \geq 3$ for some j, the number of hidden neurons N_H can be reduced. If $\beta_1 = 3$ the above Boolean function can be realized by the BNN with only one hidden neuron ($N_H = 1$): $y = \mathrm{sgn}(\xi_1 + 0)$ and $\xi_1 = \mathrm{sgn}(-x_1 - x_2 - x_3 - 0)$. The BNN may be useful for simple realization of Boolean functions and we can say " *If $\beta_j = 1$ for all j then the BNN with $M = 1$ is equivalent to the DFC. If β_j increases for some j then the number of hidden neurons (N_H) can decrease.* "

Let $M = N$. Applying the feedback with delay, we obtain the DBNN that can generate a variety of BVSs:

$$\boldsymbol{x}(t+1) = F_b(\boldsymbol{x}(t)) \text{ or } x_i(t+1) = F_{bi}(\boldsymbol{x}(t)) \tag{2}$$

where t is a discrete time and $i = 1 \sim N$. $\boldsymbol{x}(t+1)$ and $\boldsymbol{x}(t)$ correspond to $\boldsymbol{y}$ and $\boldsymbol{x}$ in Eq. (1), respectively. For convenience, we define the k-th hidden subset by $\boldsymbol{H}_k \equiv \{\xi_j | w_{kj}^o = 1\}$: the k-th output y_k is disjunction of all the hidden outputs

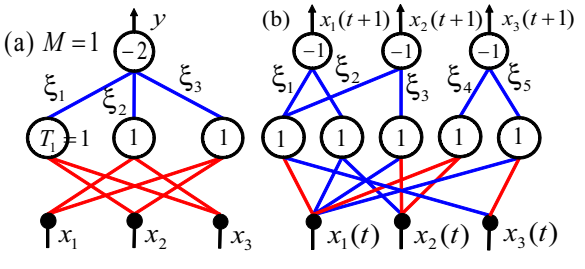


Fig. 1. BNN examples. The blue (red) branch means $w_{ij} = 1$ ($w_{ij} = -1$).

ξ_j in $\boldsymbol{H}_k$. Fig. 1 (b) shows a simple example of DBNN where the hidden subsets are $\boldsymbol{H}_1 = \{\xi_1, \xi_2\}$, $\boldsymbol{H}_2 = \{\xi_1, \xi_3\}$ and $\boldsymbol{H}_3 = \{\xi_4, \xi_5\}$. In this case, ξ_1 is shared in $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$. This DBNN generates the BVS with period 8: $\boldsymbol{x}(1)$ =(000), $\boldsymbol{x}(2)$ =(001), $\boldsymbol{x}(3)$ =(111), $\boldsymbol{x}(4)$ =(100), $\boldsymbol{x}(5)$ =(011), $\boldsymbol{x}(6)$ =(110), $\boldsymbol{x}(7)$ =(101), $\boldsymbol{x}(8)$ =(010), $\boldsymbol{x}(9) = \boldsymbol{x}(1)$ where "0" is used instead of "-1" for simplicity.

3 GA-Based Learning Algorithm

Here we define the GALA. Let $\{\boldsymbol{z}(1), \cdots, \boldsymbol{z}(N_s)\}$ ($\boldsymbol{z}(t) \in \boldsymbol{B}^N$) denote a teacher signal of BVS. Let a Boolean function $F_{ai} : \boldsymbol{B}^N \to \boldsymbol{B}$ governs the i-th elements of the BVS: $z_i(t+1) = F_{ai}(\boldsymbol{z}(t))$. If a teacher signal is given, the problem is finding the hidden neuron parameters (w_{ji}, β_j) and to construct the hidden subsets $\boldsymbol{H}_k$. First, the j-th element of the DBNN F_{bj} is subject to realize a Boolean function F_{aj}. For the Boolean function F_{aj}, an element $\boldsymbol{u}_l$ of $\boldsymbol{B}^N$ is said to be a true vertex if $F_{aj}(\boldsymbol{u}_l) = 1$. An element $\boldsymbol{v}_k$ of $\boldsymbol{B}^N$ is said to be a false vertex if $F_{ai}(\boldsymbol{v}_k) = -1$. The set of the true and false vertices are denoted by $\boldsymbol{U} = \{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{N_u}\}$ and $\boldsymbol{V} = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{N_v}\}$, respectively. $N_u + N_v \leq 2^N$ is satisfied. Elements in neither $\boldsymbol{U}$ nor $\boldsymbol{V}$ are referred to as "don't care". Note that the j-th hidden neuron corresponds to the j-th separating hyper plane (SHP): $\sum_{i=1}^{N} w_{ji}x_i - T_j = 0$. Fig. 2 shows SHPs corresponding to DBNN in Fig. 1 (b). The GALA tries to determine the parameters (w_{ji}, β_j) using the true vertices in the teacher signal: $\{\boldsymbol{z}(t) | F_{ai}(\boldsymbol{z}(t)) = 1,\ i = 1 \sim N\}$. The following 4 steps are repeated for F_{ai} from i=1 to N.

Step 1: (initialization). Let j be the index of hidden neuron and let $j = 1$.
Step 2: Applying the GA-subroutine defined afterward, the j-th SHP is decided. The j-th hidden output ξ_j is added to the i-th hidden subset $\boldsymbol{H}_i$.
Step 3 (SHP sharing): Apply the j-th SHP to separation of true vertices of the other outputs: F_{ak} for $k = i \sim N$. If it can be used for the k-th output (as SHP_1 in Fig. 2) then ξ_j is added to the k-th hidden subset $\boldsymbol{H}_k$. All the separated true vertices are declared as "don't care".
Step 4: Let $j = j + 1$, go to Step 2 and repeat until all the true vertices are separated.

GA-subroutine: The fitness is the number of separated true vertices. The Mg pieces of chromosomes $\{\boldsymbol{C_1}, \cdots, \boldsymbol{C_{Mg}}\}$ are candidates of the weights w_{ji} of the j-th SHP: $\boldsymbol{C_k} = (w_{j1}^k, \cdots, w_{jN}^k)$. One of the initial chromosome is selected from true vertices of teacher signals (e.g., $\boldsymbol{C}_1 = \boldsymbol{z}(1)$). This chromosome is said to be the initial kernel. It guarantees separation of at least one true vertex and is

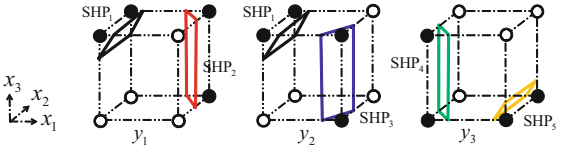


Fig. 2. SHPs. Black and white circles denote true and false vertices, respectively.

Table 1. Teacher signal BVS1 ($\boldsymbol{z}(t+14) = \boldsymbol{z}(t)$), $N_s = 15$, "0" means "-1")

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
z_1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1
z_2	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0
z_3	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
z_4	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0
z_5	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1
z_6	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1
z_7	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1

Table 2. Teacher signal BVS2 ($\boldsymbol{z}(t+15) = \boldsymbol{z}(t)$), $N_s = 16$, "0" means "-1")

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
z_1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0
z_2	0	0	0	1	1	0	0	0	1	1	1	1	1	1	1	0
z_3	0	0	1	1	1	0	0	1	1	0	0	1	1	1	1	0
z_4	0	1	1	1	0	0	1	1	1	0	1	1	0	1	1	0
z_5	1	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1
z_6	1	1	0	0	0	1	1	0	0	1	1	0	1	1	1	1
z_7	1	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1

evolved to separate true vertices as many as possible. Other chromosomes are set randomly. For each $\boldsymbol{C_k}$, the parameter $\beta_j \geq 1$ is selected to give the best fitness value. The M_g pieces of chromosomes are selected to the next generation by G_{max} times repeating of the elite strategy and ranking selection. The two-point crossover is applied with probability P_c. In the mutation, one gene w^l_{ji} is selected with probability P_m and the value is reset to either of $\{-1, 0, 1\}$.

4 Numerical Experiments

Here the GALA is tired to store two teacher signals: BVS1 and BVS2 in Tables 1 and 2. The parameters are fixed after trial-and-errors: $M_g = 10$, $P_c = 0.2$, $P_m = 0.1$, and $G_{max} = 30$. The BVS1 is the 7-dimensional periodic BVS with period 14 that relates to the multi-phase PWM control signal in dc-ac inverters [8] [9]. Applying the GALA, we obtain the DBNN in Fig. 3 (a) in which the BVS1 is stored successfully. Table 3 shows the parameters after the learning. The initial kernel and $\beta_j \geq 3$ are effective to reduce N_H to 7 ($N_H = 11$ in Ref. [10]). The hidden sets are $\boldsymbol{H}_k = \{\xi_k\}$ for $k = 1 \sim N$: no SHP sharing. Then we have confirmed convergence to BVS1 from all the initial values. Fig. 4 shows an example of the pull-in process. *The storing function of the GALA is guaranteed and the stabilization function is unclear, however, the stored BVS1 is stabilized (and so is BVS2).* Although this is an experimental fact, it can be a trigger to develop novel learning algorithms with stabilization function. As β_j increases, the SHP$_j$ can evolve to separate larger number of true vertices and may contribute the stabilization, however, its theoretical analysis is in progress.

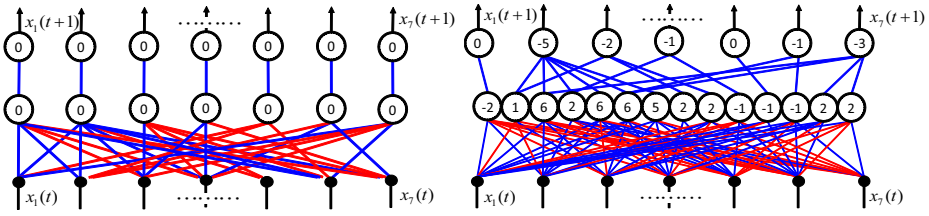


Fig. 3. DBNNs for BVS1 (a) and BVS2 (b). The blue (red) branch means $w_{ij} = 1$ ($w_{ij} = -1$).

Table 3. Parameters for BVS1 after the learning

j	w_{j1}	w_{j2}	w_{j3}	w_{j4}	w_{j5}	w_{j6}	w_{j7}	β_j
1	1	1	-1	-1	1	-1	1	7
2	1	1	1	1	-1	1	1	7
3	0	0	1	-1	-1	-1	-1	5
4	0	0	1	1	0	0	-1	3
5	-1	-1	0	0	0	-1	0	3
6	0	-1	0	1	0	0	1	3
7	1	-1	-1	-1	0	1	0	5

Table 4. Parameters for BVS2 after the learning

j	w_{j1}	w_{j2}	w_{j3}	w_{j4}	w_{j5}	w_{j6}	w_{j7}	β_j
1	1	1	-1	-1	-1	-1	-1	9
2	-1	-1	1	0	0	-1	0	3
3	1	1	1	1	-1	-1	-1	1
4	0	1	0	-1	1	-1	1	3
5	1	1	-1	1	1	1	-1	1
6	1	1	1	1	1	-1	-1	1
7	1	0	1	1	1	1	-1	1
8	0	0	0	1	1	0	-1	1
9	-1	1	1	-1	0	1	0	3
10	-1	-1	-1	0	1	1	-1	7
11	1	1	-1	0	0	1	0	5
12	1	0	1	-1	-1	1	1	7
13	1	0	1	1	-1	1	0	3
14	1	0	0	-1	-1	-1	1	3

The second periodic teacher signal BVS2 is constructed artificially. Applying the GALA, we obtain the DBNN as shown in Fig. 3 (b): $N_H = 14$ and 4 SHPs are shared. The 4 SHPs correspond to 4 hidden outputs ξ_2, ξ_3, ξ_4 and ξ_6. Table 4 shows the parameters including $\beta_j \geq 3$. The stabilization function for all the initial values is confirmed also in this example.

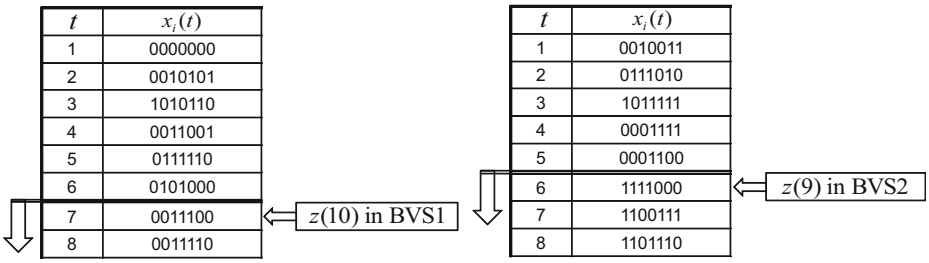


Fig. 4. Pull-in process to the stored BVSs ("0" means "-1")

5 Conclusions

The DBNN and GALA have been studied in this paper. The GALA includes effective initial kernel of chromosome and SHP sharing. In basic numerical experiments, we have confirmed that the GALA can store desired periodic BVSs and the stored BVSs are stabilized. Future problems are many, including analysis of the learning process, analysis of the stabilization function, storing larger BVSs and engineering applications.

References

1. Gray, D.L., Michel, A.N.: A training algorithm for binary feed forward neural networks. IEEE Trans. Neural Networks 3(2), 176–194 (1992)
2. Kim, J.H., Park, S.K.: The geometrical learning of binary neural networks. IEEE Trans. Neural Networks 6(1), 237–247 (1995)
3. Muselli, M., Liberati, D.: Training Digital Circuits with Hamming Clustering. IEEE Trans. Circuits Syst. I 47(4), 513–527 (2000)
4. Yamamoto, A., Saito, T.: A flexible learning algorithm for binary neural networks. IEICE Trans. Fundamentals E81-A(9), 1925–1930 (1998)
5. Chen, F., Chen, G., He, Q., He, G., Xu, X.: Universal perceptron and DNA-like learning algorithm for binary neural networks: non-LSBF implementation. IEEE Trans. Neural Networks 20(8), 1293–1301 (2009)
6. Wada, W., Kuroiwa, J., Nara, S.: Errorless reproduction of given pattern dynamics by means of cellular automata. Phys. Rev. E 68(036707), 1–8 (2003)
7. Kabeya, S., Saito, T.: A GA-based flexible learning algorithm with error tolerance for digital binary neural networks. In: Proc. IEEE-INNS Joint Conf. Neural Netw., pp. 1476–1480 (2009)
8. Boost, M.A., Zipgas, P.D.: State-of-the-art carrier PWM techniques: a critical evaluation. IEEE Trans. Ind. Applicat. 24, 271–280 (1988)
9. Bose, B.K.: Neural network applications in power electronics and motor drives - an introduction and perspective. IEEE Trans. Ind. Electron. 54(1), 14–33 (2007)
10. Ito, R., Saito, T.: Dynamic binary neural networks and evolutionary learning. In: Proc. IEEE-INNS Joint Conf. Neural Netw., pp. 1683–1687 (2010)
11. Kim, K.-J., Cho, S.-B.: Evaluation of Distance Measures for Speciated Evolutionary Neural Networks in Pattern Classification Problems. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009. LNCS, vol. 5864, pp. 630–637. Springer, Heidelberg (2009)

Self-organizing Digital Spike Interval Maps

Takashi Ogawa and Toshimichi Saito*

Hosei University, Koganei, Tokyo, 184-8584 Japan
http://www.hosei.ac.jp

Abstract. This paper studies digital spike interval maps and its learning algorithm. The map can output a variety of digital spike-trains. In order to learn a desired spike-train, two maps are switched by the contradiction detector and they evolve with self-organizing and growing functions. Performing basic numerical experiments for two examples, algorithm efficiency is confirmed.

Keywords: Spiking neurons, inter-spike-interval, supervised learning.

1 Introduction

Spike signals play important roles in various artificial/biological neural systems. Analysis of them is basic to understand spike-based nonlinear dynamics and information processing function in the brain [1]-[6]. Several spiking neuron models have been presented in order to analyze rich chaotic/periodic spike-trains and related bifurcation phenomena [1]-[5]. Applying pulse-coupling to the spiking neurons, we obtain pulse-coupled neural networks having rich synchronous/asynchronous phenomena. The spike signals are simple, low power and suitable for various real/potential applications: image segmentation [7], spike-based communications [8], neural prosthesis [9], etc. Learning and realization of desired spike-trains are important.

This paper presents digital spike interval map (DSIM) and its learning algorithm. The domain of the DSIM is a set of lattice points. The DSIM outputs a sequence of digital inter-spike intervals (ISI) that can be converted to a digital spike-train (DST). The ISI is robust for time-delay in the transmission and is basic to consider spike-based encoding. The DSIM outputs a variety of DSTs and the richness is suitable to learn a variety of teacher signals. Our learning algorithm is based on two DSIMs switched by a contradiction detector (CTD). In order to learn a desired DST, the CTD switches two DSIMs and the switched DSIMs evolve with self-organizing [10] and growing functions. We have performed numerical experiments for two examples: a DST from two bifurcating neurons (BN [3] [4]) and a DST from hyperchaotic spiking oscillator (HCSO [5]). In these experiments, the algorithm efficiency is confirmed.

Note that the switched DSIMs can be regarded as a developed version of basic digital spike maps (DSM [11] [12]) and can learn wider class of DSTs than the DSM. Also, the DSIM and DSM are simple digital dynamical systems with rich phenomena such as cellular automata ([12] and references therein).

* This work is supported in part by JSPS KAKENHI#21500223.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 612–617, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 The Digital Spike Interval Map and Learning

The domain of the DSIM is a set of lattice points $I_D \equiv \{l_1, l_2, \cdots, l_M\}$ where $l_i = (2i-1)/(2M)$ and $i = 1 \sim M$. The DSIM has K particles $\{a_1, \cdots, a_K\}$. The j-th particle α_j is characterized by its x- and y-coordinates: $\alpha_j = (x_j,\ y_j)$, $x_j \in I_D$ and $y_j \in I_D$. As shown in Fig. 1 (a), each particle connects to the both side particles and forms a ring topology: $\alpha_{j+K} \equiv \alpha_j$ for $j = 1 \sim K$. In the learning, the number of particles K can increase at most M. The DSIM is a mapping from I_D to itself as shown in Fig. 1:

$$y_c = Q(x), \quad \text{where } \alpha_c = (x_c, y_c),\ |x_c - x| = \min_i |x_i - x| \tag{1}$$

If an input $x \in I_D$ is given then the DSIM finds the winner particle $\alpha_c = (x_c, y_c)$ whose x-coordinate x_c is the closest to x. The y-coordinate y_c becomes the output of Q. Iterating the mapping Q for an initial spike interval φ_1, we obtain a DST= $\{\tau_1, \cdots, \tau_N\}$: $\tau_n = \sum_{i=1}^{n} \varphi_i$, $\varphi_{n+1} = Q(\varphi_n)$ where $n = 1 \sim N$ and N is the number of spike-positions. The DST is represented by its spike-positions. Hereafter φ_k is referred to as a digital ISI. As the particle arrangement varies, the DSIM can exhibit rich phenomena as is the quantized return map [4]. This richness is convenient for learning of DSTs that is the object of this paper. The teacher signal is based on a DST:

$$\{\tau_1, \cdots, \tau_N\},\ \tau_n = \theta_1 + \cdots + \theta_n,\ \ \text{ISI: } \theta_n \in I_D,\ n = 1 \sim N,\ N < M. \tag{2}$$

If an analog spike-train is given, the DST is given by a quantization as shown in the Section 3. Let s denote a discrete learning time. Let $\beta(s) = (\theta_s, \theta_{s+1})$ denote a pair of digital ISIs. The teacher signal is $\{\beta(1), \cdots, \beta(N-1)\}$: $\beta(1)$ is presented at $s = 1$, $\beta(2)$ is presented at $s = 2$, $\cdots$, and (θ_{N-1}, θ_N) is presented at $s = N - 1 \equiv s_{max}$. In the learning, we prepare DSIM1 and DSIM2 either of which can be active (Fig. 1 (b)). Each particle can be either free or fixed. All the particles are free at $s = 0$. A particle is fixed after it moves to the position of a teacher signal as defined below. In the following learning algorithm, we try to change the position and number of particles to approximate the DST.

Step 1: Let $s = 1$. $K(s)$ particles are assigned in equidistant on L_D as shown in Fig. 2 (a). The CTD activates either DSIM1 or DSIM2.

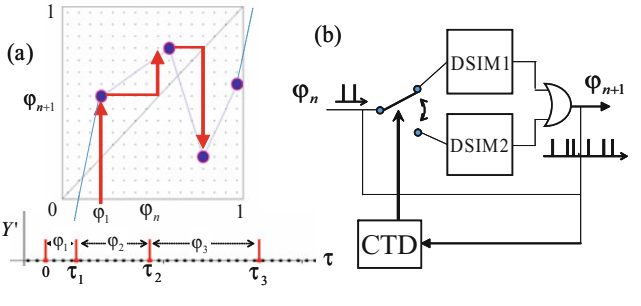


Fig. 1. The DSIM (a) and activation by contradiction detector (CTD)

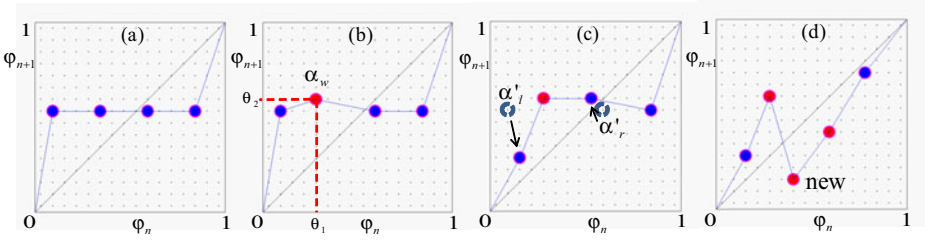


Fig. 2. Learning process for DSIM (a) initialization, (b) update of winner, (c) update of neighbor, (d) birth of a new particle. Blue (red) circles denote free (fixed) particles.

Step 2: The teacher signal $\beta(s) = (\theta_s, \theta_{s+1})$ is presented at time s.

Step 3: The active DSIM finds the right- and left-closest particles $\alpha_r = (x_r, y_r)$ and $\alpha_l = (x_l, y_l)$ where $0 \leq x_r - \theta_s < x_i - \theta_s$ and $0 < \theta_s - x_l < \theta_s - x_i$ for $i = 1 \sim K(s)$. We consider two cases.

Case 1: $\theta_s \neq x_r$. If both α_r and α_l are free then the closer particle becomes the winner: α_r becomes the winner if $|x_r - \theta_s| \leq |\theta_s - x_l|$ and α_l becomes the winner if $|x_r - \theta_s| > |\theta_s - x_l|$. If either α_r or α_l is free then the free particle becomes the winner. If the winner is determined then go to Step 4. If both α_r and α_l are fixed then go to Step 5.

Case 2: $\theta_s = x_r$. If α_r is free then α_r becomes the winner and go to Step 4. If α_r is fixed and $y_r = \theta_{s+1}$ then go to Step 6. If α_r is fixed and $y_r \neq \theta_{s+1}$ then $\beta(s)$ is declared to be *contradictive*. In this case the CTD activates the other DSIM as shown in Fig. 1 (b)) and go to Step 3. If the $\beta(s)$ is contradictive also for the activated DSIM then $\beta(s)$ is declared to be *double-contradictive* and go to Step 5 (i.e., the double-contradictive $\beta(s)$ is ignored).

Step 4: Let $\alpha_w = (x_w, y_w)$ denote the winner particle. The α_w is updated to the position of the teacher signal : $\alpha_w = \beta(s)$ as shown in Fig. 2 (b). α_w is fixed hereafter. If the right-closest particle $\alpha'_r = (x'_r, y'_r)$ of α_w is free then x'_r approaches by one lattice point to the winner and y'_r moves on the line connecting the winner and the left particle of α'_r (Fig. 2 (c)). In a likewise manner the left-closest particle of α_w is interpolated. Go to Step 6.

Step 5: If no lattice point exists between α_r and α_l then go to Step 6. If some lattice point exists between α_r and α_l then a new particle α_{new} is born and is inserted in the position of the teacher signal with connection to the both sides (Fig. 2 (d)). Let $K = K + 1$. The new particle is fixed hereafter.

Step 6: Let $s = s + 1$, go to Step 2 and repeat until the time limit s_{max}.

Remark 1: The switched DSIMs can learn the teacher signal if $\beta(s)$ is not double-contradictive for $1 \leq s \leq s_{max}$ (even if it is contradictive). The single DSIM (and DSM) can not learn the contradictive signals but can learn non-contractive signals such as simple periodic DSTs. Update of the winner and neighbors in Step. 4 and birth of a particle in Step 5 refer to the (growing) SOMs [10] [13].

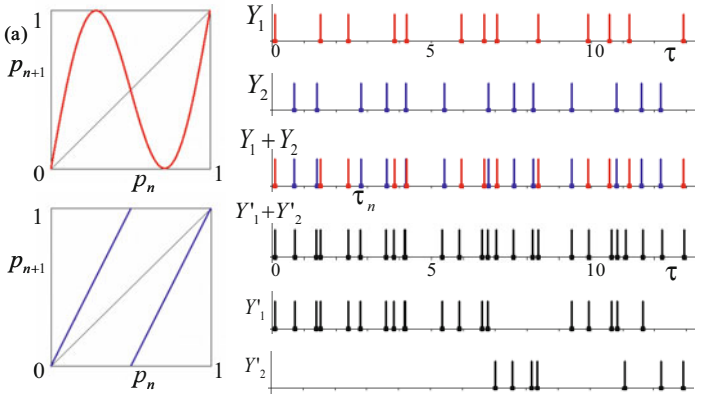


Fig. 3. The first example. (a) Basic spike-phase maps ($p_n = t_n \bmod 1$) Y_1: DST of F_1 for $t_1 = 0$, Y_2: DST of F_2 for $t_1 = 0.7$, $Y_1 + Y_2$: DST for teacher signal, Y_1': DST of DSIM1 after the learning, Y_2': DST of DSIM2 after the learning.

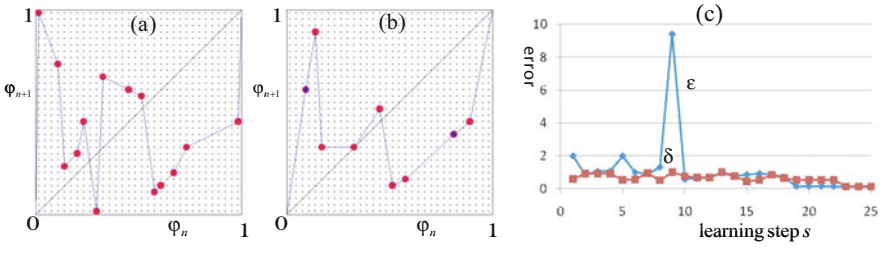


Fig. 4. Learning results. DSIM1 (a) and DSIM2 (b) after the learning where red (blue) circles denote fixed (free) particles. (c) Approximation error in the learning process.

3 Numerical Experiments

We apply the algorithm to two examples of teacher signals. The first one is based on two BNs (BN1 and BN2) whose spike-phase maps are

$$\begin{aligned} &\text{BN1: } p_{n+1} = p_n - b_1(p_n) \equiv F_1(p_n), \;\; b_1(t) = -k \sin 2\pi t \\ &\text{BN2: } p_{n+1} = p_n - b_2(p_n) \equiv F_2(p_n), \;\; b_2(t) = -\alpha t \text{ for } |t| < 0.5 \end{aligned} \tag{3}$$

where $b_2(t+1) = b_2(t)$, $0 < k < 1$ and $0 < \alpha \leq 1$. $b_1(t)$ and $b_2(t)$ correspond to base signals. Using the spike phase $p_n \in [0, 1)$, the n-th spike position is given by $t_n = p_n + (n-1)$. Derivation of the maps can be found in [4]. For simplicity, we fix the parameters $k = 0.73$ and $\alpha = 1$ which give the map in Fig. 3 (a). Extracting ISIs form the spike-trains and quantizing each ISI onto the closest lattice point in I_D, we obtain DSTs Y_1 and Y_2 as shown in Fig. 4. Adding these two DSTs ($Y_1 + Y_2$) and extracting ISIs, we obtain the teacher signal DST (Eq. (2)) where the ISI is normalized by its maximum value $\Delta t_{max} = 1.18$. Let N_1

and N_2 be the number of spikes of Y_1 and Y_2, respectively. We set $N_1 = 13$, $N_2 = 13$, # lattice points $M = 32$ and # initial particles $K(0) = 8$ for DSIM1 and DSIM2. Applying the algorithm, we obtain DSIMs in Fig. 4 and DSTs in Fig. 3: the DSIMs can approximate the teacher signal. In Fig. 3 the DSIM1 is active until 14 spikes, a contradict input causes switching to DSIM2, and the activation switching is repeated 5 times. This switching is effective to approximate $Y_1 + Y_2$. Fig. 4 (c) shows the error characteristics in the learning process where we have measured difference between the teacher signal and DTS by the maximum ISI error $\delta = \max_n |\theta_n - \varphi_n|$ and the maximum DST error $\varepsilon = \max_n |\tau_n - \tau'_n|$. The peak of ε is due to the contradictive signal. This teacher signal is based on two 1D maps, is not double-contradictive and the learning is completed at $s = s_{max}$.

Remark 2: Single DSIM (and DSM) is hard to learn this teacher signal because it is not based on single 1D map and can be contradictive. The error becomes sufficiently small before s_{max} (Fig. 4 (c)). The DSIMs can have free particles and can output different DTSs from the teacher signal.

The second teacher signal is based on the HCSO described by

$$\begin{array}{ll} \dot{x}_1 = x_3,\ \dot{x}_2 = \gamma_1(x_2 - x_3),\ \dot{x}_3 = \gamma_2(x_2 - x_1) & \text{for } x_1 < 1 \\ (x_1(\tau+), x_2(\tau+), x_3(\tau+)) = (q, x_2(\tau), x_3(\tau)) & \text{if } x_1(\tau) = 1 \end{array} \tag{4}$$

where $\dot{x} \equiv dx/d\tau$. When x_1 is reset to q the HCSO outputs a spike. Fig. 5 (a) shows hyperchaotic attractor for $\gamma_1 = 1$, $\gamma_2 = 70$ and $q = 0.6$. In Ref. [5], the attractor has been observed in a simple circuit and the dynamics is integrated into a 2D map in which positiveness of two Lyapunov exponents is confirmed. Applying the quantization, we obtain the DST Y that is the object of learning. Let $N = 19$, $M = 32$ and let $K(0) = 8$. The ISI is normalized by $\Delta t_{max} = 1.07$. Applying the algorithm, the switched DSIMs can approximate the teacher signal and the error can be small before $s = s_{max}$ as shown in Fig. 5 and 6. The system repeats the activation switching 7 times.

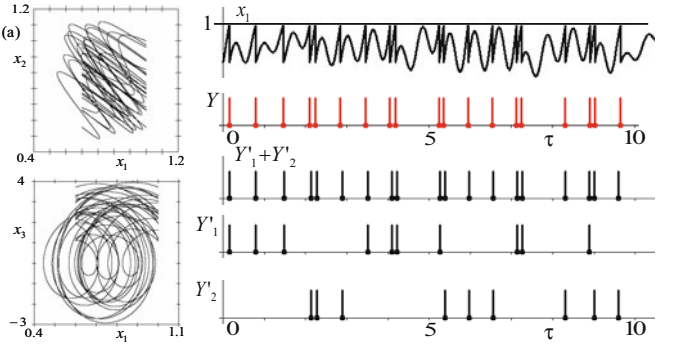


Fig. 5. The second example. (a) Hyperchaotic attractor Y: DST for teacher signal, Y'_1: DST of DSIM1 after the learning, Y'_2: DST of DSIM2 after the learning.

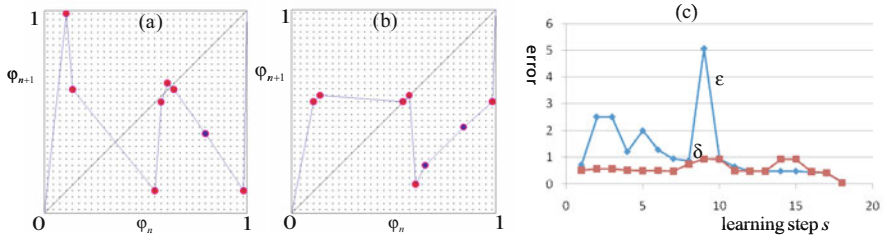


Fig. 6. DSIM1 (a) and DSIM2 (b) after the learning. (c) Approximation error.

4 Conclusions

The DSIM and its learning algorithm have been studied in this paper. The algorithm includes a self-organizing function and growing structure: they are suitable to learn a variety of spike-trains. The algorithm efficiency is confirmed in numerical experiments for two examples based on the BNs and HCSO. Future problems include analysis of learning process, analysis of approximation properties, development into to multi-coupled DSIMs, and engineering applications.

References

1. Izhikevich, E.M.: Simple Model of Spiking Neurons. IEEE Trans. Neural Networks 14(6), 1569–1572 (2003)
2. Toyoizumi, T., Aihara, K., Amari, S.: Fisher information for spike-based population coding. Phys. Rev. Lett. 97, 098102 (2006)
3. Perez, R., Glass, L.: Bistability, period doubling bifurcations and chaos in a periodically forced oscillator. Phys. Lett. 90A(9), 441–443 (1982)
4. Torikai, H., Saito, T.: Return map quantization from an integrate-and-fire model with two periodic inputs. IEICE Trans. Fund. E82-A(7), 1336–1343 (1999)
5. Takahashi, Y., Nakano, H., Saito, T.: A simple hyperchaos generator based on impulsive switching. IEEE Trans. Circuits Syst. II 51(9), 468–472 (2004)
6. Hashimoto, S., Torikai, H.: A novel hybrid spiking neuron: bifurcations, responses, and on-chip learning. IEEE Trans. Circuits Syst. I 57(8), 2168–2181 (2010)
7. Campbell, S.R., Wang, D., Jayaprakash, C.: Synchrony and desynchrony in integrate-and-fire oscillators. Neural Computation 11, 1595–1619 (1999)
8. Sushchik, M., Rulkov, N., Larson, L., Tsimring, L., Abarbanel, H., Yao, K., Volkovskii, A.: Chaotic pulse position modulation: a robust method of communicating with chaos. IEEE Comm. Lett. 4, 128–130 (2000)
9. Torikai, T., Nishigami, T.: An artificial chaotic spiking neuron inspired by spiral ganglion cell. Neural Networks 22, 664–673 (2009)
10. Kohonen, T.: Self-organizing maps. Springer, Heidelberg (2001)
11. Torikai, H., Funew, A., Saito, T.: Digital spiking neuron and its learning for approximation of various spike-trains. Neural Networks 21, 140–149 (2008)
12. Ogawa, T., Saito, T.: Digital spike maps and learning of spike signals. In: Proc. IEEE-INNS Int'l. Joint Conf. Neural Netw., pp. 1587–1592 (2010)
13. Oshime, T., Saito, T., Torikai, H.: ART-Based Parallel Learning of Growing SOMs and its Application to TSP. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) ICONIP 2006. LNCS, vol. 4232, pp. 1004–1011. Springer, Heidelberg (2006)

Shape Space Estimation by SOM^2

Sho Yakushiji and Tetsuo Furukawa

Kyushu Institute of Technology, Kitakyushu 808-0196, Japan
yakushiji-sho@edu.brain.kyutech.ac.jp,
furukawa@brain.kyutech.ac.jp
http://www.brain.kyutech.ac.jp/~furukawa

Abstract. This study aims to develop an estimation method for a shape space. In this work, 'shape space' is a nonlinear subspace formed by a class of visual shapes, in which the continuous change in shapes is naturally represented. By estimating the shape space, various operations dealing with shapes, such as identification, classification, recognition, and interpolation can be carried out in the shape space. A higher-rank of self-organizing map (SOM^2) is employed as an implementation of the shape space estimation method. Simulation results show the capabilities of the method.

Keywords: shape representation, shape space, self-organizing map, higher-rank of SOM.

1 Introduction

The shape of an object in a visual scene is one of the most important clues for recognizing and identifying the object. Indeed shape information is used in human identification in natural scenes, automatic processing of medical images, online search of trademarks, pose recognition of humans, and so on. In addition, some types of hand-written character recognition can also be regarded as a kind of shape recognition task [1,2,3,4].

Roughly speaking, there are two different approaches to shape classification or recognition, namely, *the shape description* and *the shape representation* approaches [5,6]. In *the shape description* approach, shape features such as roundness or eccentricity are quantified, and then described as a feature vector. One of the advantages of this approach is that it is fairly easy to design the classifier if the shape feature vector has been designed appropriately so as to describe the essential features relevant to the task. It is also easy to ignore observation dependent transformations, such as rotation, translation, and scale changes in the visual image. This means that the most important issue for this approach is how to design the shape feature vector. Sometimes this is difficult, since it is highly dependent on the task. In addition, information of the shape itself is lost, and the original shape image cannot be restored from the shape feature vector. This issue is known as the pre-image problem [7].

By contrast, in *the shape representation* approach, the shape information is preserved and each shape image is represented by a shape model, such as a contour manifold or a skeleton graph. One of the typical methods represents the shape contour using a closed manifold or a periodical function [6]. Another example of the shape representation method uses a self-organizing map (SOM) or one of its modifications, in which

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 618–627, 2011.
© Springer-Verlag Berlin Heidelberg 2011

the contour or the skeleton is represented by a graph structure consisting of SOM units [8,9]. In this case, concatenation of the reference vectors is regarded as the shape model. Since shape information is preserved in this approach, there is no chance of encountering the pre-image problem. Not only is it possible to restore the given shape images, but also to generate intermediate shape images between the given images. Therefore, the shape representation approach provides more general methods for dealing with shapes. Notwithstanding its advantages, the computational cost of this approach is usually more expensive than that of the shape description approach. Furthermore, this approach is often affected by observation transformation.

The aim of this work is to develop a method for estimating *the shape space* using the shape representation approach. Here the word '*shape space*' means a continuous space consisting of a set of shapes belonging to the same category. For example, shapes of leaves of the same species forms a shape space of leaves. Similarly, the shape of the handwritten letter 'A' varies within the shape space of the letter 'A', which consists of all variations of the letter. It is expected that estimating such a shape space will facilitate dealing with shapes, e.g., identification, classification, quantization, interpolation, and so on. Thus our goal is to develop a method for estimating such a shape space from a finite number of given shapes, even though the shape space may consist of an infinite number of shape variations.

The key idea is to use a higher-rank of self-organizing map, called a SOM^n [10]. The second rank of SOM^n, i.e., SOM^2, has a hierarchical structure consisting of a set of lower (1st) SOMs and a higher (2nd) SOM. The task of the SOM^2 is to organize a map of the set of self-organizing maps organized at the lower level. Thus, if two 1st maps are similar to each other, then they are mapped into neighboring locations in the 2nd map, whereas distinctly different 1st maps are located further apart in the 2nd map. As a result, the entire SOM^2 represents the data distributions in a product space called a fiber bundle. Since a SOM could be employed to represent an object shape, the SOM^2 would have the ability to generate a map of shapes. In this paper, the concept of shape space estimation is first introduced, and then the algorithm based on the SOM^2 is described. Finally, some simulation results are presented.

2 Framework

2.1 The Generative Model and Goal of the Task

First, let us clarify our goal even further. Like other modern machine learning works, the data generative model is assumed in our framework as well. Fig.1 shows the generative model of shapes used in this work. The latent shape space determines the essential property of the shapes belonging to a certain class (e.g., leaves or letters). Thus, shapes belonging to the same shape space are all isomorphic. The positional vector in the latent shape space is represented by $\omega \in \Omega$. Every shape belonging to the class $s \in \mathbf{S}$ corresponds uniquely to a point in the latent shape space $\omega(s)$, and vice versa. Furthermore, two neighboring points in the shape space correspond to two similar shapes. If a shape is moved continuously from ω_1 to ω_2 in the shape space, a morphing shape from s_1 to s_2 is continuously observed. In the case of letter 'A', one can observe a continuous change in the handwritten letter 'A' from one to another while keeping the shape of the

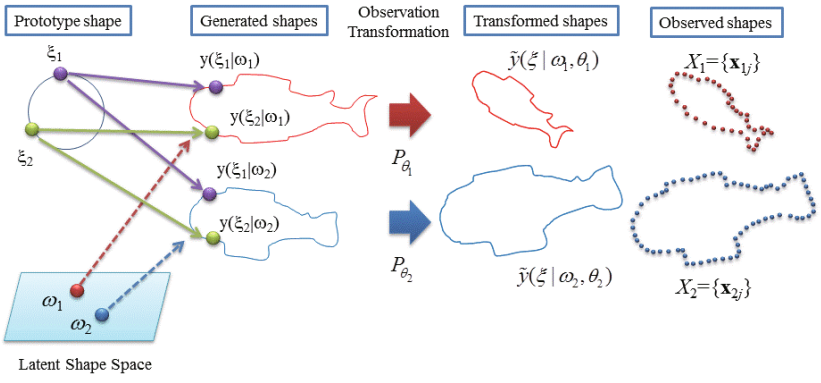


Fig. 1. The generative model of shapes used in this work. The goal is to estimate the latent shape space.

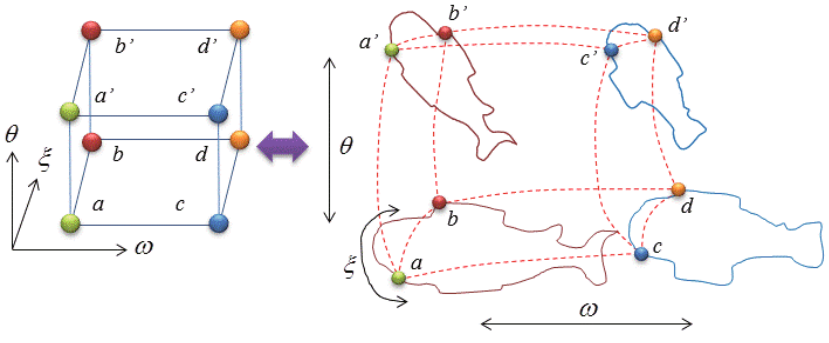


Fig. 2. The product latent space of ξ, ω, and θ is mapped onto the observed shape space. The four points labeled as a, b, c, and d indicate how the mapping is carried out.

letter 'A'. The shape of letter 'B' never appears, since this shape does not belong to the shape space of 'A'. Therefore, the shape space is in fact a nonlinear subspace consisting of a class of shape models. This is the reason that the shape representation approach is required in this work.

Once a latent variable ω has been generated, it is mapped to the shape function $\mathbf{y}(\xi \mid \omega)$, which generates an actual shape from the prototype (Fig. 1). Thus,

$$\begin{aligned} \Phi : \Omega &\longrightarrow \mathcal{Y} \\ \omega &\longmapsto \mathbf{y}(\xi \mid \omega). \end{aligned}$$

Here, ξ represents the position in the prototype, and $\mathbf{y}(\xi \mid \omega)$ is the corresponding point in the actual shape. Therefore, a class of shapes is assumed to be represented by a homotopy $\mathbf{y}(\xi \mid \omega)$. If the prototype is a unit circle, then ξ represents a position along the circumference, and various shapes that are isomorphic to the unit circle can be generated. After each shape is generated, it undergoes observation transformation P. Since P varies for each observation, P is supposed to be randomly chosen from the set of

observation transformations $\mathcal{P}$. Here the members of $\mathcal{P}$ are assumed to be parameterized by $\boldsymbol{\theta}$, so that the transformed shape is represented by $\tilde{\mathbf{y}}(\boldsymbol{\xi} \mid \omega, \boldsymbol{\theta}) = P_{\boldsymbol{\theta}}\, \mathbf{y}(\boldsymbol{\xi} \mid \omega)$. To be more precise, $\mathcal{P}$ is required to be a group. In this paper, we deal mainly with cases of the Lie group, in which $P_{\boldsymbol{\theta}}$ is differentiable by $\boldsymbol{\theta}$. The affine deformation set is a typical example of a Lie group.

After the transformed shape is generated, a set of contour pixels X is observed. Since the observation transformation varies for each observation, it often happens that two similar shapes produce quite different observed images.

Under this framework, our goal is to estimate the latent shape space from a finite set of shapes. More precisely, the task of the proposed method is to estimate the homotopy $\tilde{\mathbf{y}}(\boldsymbol{\xi} \mid \omega, \boldsymbol{\theta})$ and the latent variables $\boldsymbol{\xi}$, ω, $\boldsymbol{\theta}$ from the observed shapes (Fig. 2). After the shape space has been estimated, various operations such as classification, recognition, etc., can be executed in the shape space. Although the existence of such a generative model and the latent space is a working assumption, it is quite convenient for clearly defining the task. Thus, unlike conventional classification works, our main purpose is *not* classification itself, but discovering a general shape model representing the given shape class.

2.2 Training Data

Here, let us suppose that the training data are encoded in the simplest way, because it is convenient to outline the generalized case. Thus, the object in the image is separated from the ground, the border pixels are extracted, and we obtain a set of border pixels $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_J\}$. Here each $\mathbf{x}_j \in \mathbb{R}^D$ represents the coordinates of the j-th border pixel in the image. D is the dimension of the shape images, and usually $D = 2$. Since X is an ordinary set of vectors, the order of the members (i.e., the index j) is meaningless. Therefore, a continuous contour curve cannot be obtained from the order of j. This encoding is referred to as the dot distribution representation (DDR) in this paper. Some additional features such as the local curvature can be tacked on, but this is not discussed in this paper.

Now suppose that we have I images for training, and that they are encoded by DDR. Thus we have a family of sets $\{X_1, \ldots, X_I\}$, each of which consists of $X_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{iJ_i}\}$. Note that the number of border pixels varies depending on the image. This is the data that we can use for the latent shape space estimation.

3 Theory and Algorithm

3.1 Distance between Shapes

Considering the generative model shown in Fig. 1, the distance between two shapes can be defined by the distance between the shape functions:

$$L^2(s_1, s_2) \equiv \int \|\mathbf{y}_1(\boldsymbol{\xi}) - \mathbf{y}_2(\boldsymbol{\xi})\|^2 \, d\boldsymbol{\xi}. \tag{1}$$

Here, $\mathbf{y}_1$ and $\mathbf{y}_2$ are the shape functions of shapes s_1 and s_2, respectively. This definition denotes the total distance traversed by the contour pixels if the shape is morphed from s_1 to s_2. If the prototype is quantized to N discrete representative points, then it becomes

$$L^2(s_1, s_2) \simeq \|\mathbf{Y}_1 - \mathbf{Y}_2\|^2 = \sum_{n=1}^{N} \left\|\mathbf{y}_1(\boldsymbol{\xi}_n) - \mathbf{y}_2(\boldsymbol{\xi}_n)\right\|^2 . \tag{2}$$

Here, $\mathbf{Y}_i$ is the concatenated vector of $\mathbf{y}_i(\boldsymbol{\xi}_1), \ldots, \mathbf{y}_i(\boldsymbol{\xi}_N)$. In the shape representation approach, vector $\mathbf{Y}$ can be regarded as the shape model. This is a very natural definition from the perspective of the generative model, since $\{\mathbf{Y}\}$ forms a distance space of shapes. Similarly, the transformed shape model $\tilde{\mathbf{Y}}$ is defined as $\tilde{\mathbf{Y}} \equiv P_\theta \mathbf{Y} = (P_\theta \mathbf{y}(\boldsymbol{\xi}_n))$. Then the distance between two observed shape models is given by

$$L(\tilde{s}_1, \tilde{s}_2) = \left\|P_{\theta_1}^{-1}\tilde{\mathbf{Y}}_1 - P_{\theta_2}^{-1}\tilde{\mathbf{Y}}_2\right\| . \tag{3}$$

However, to apply this distance measure to actual tasks, we need P_{θ_1} and P_{θ_2}, which are usually unknown. Hence the following alternative distance measure between two models is introduced.

$$\tilde{L}(\tilde{\mathbf{Y}}_1 \,\|\, \tilde{\mathbf{Y}}_2) \equiv \min_{P \in \mathcal{P}} \left\|\tilde{\mathbf{Y}}_1 - P_\theta \tilde{\mathbf{Y}}_2\right\| \tag{4}$$

Since it is assumed that $\mathcal{P}$ is a group, the existence of inverse and identical transformations is assured. If $\mathcal{P}$ is a Lie group such as an affine deformation, the Newton method can be employed to find the optimum P_θ.

In the above explanation, the latent variable $\boldsymbol{\xi}$ is assumed to be known. In reality, $\boldsymbol{\xi}$ should be estimated together with the shape model $\mathbf{Y}$. Since $\boldsymbol{\xi}$ determines the correspondence between two different shapes, the estimation of $\boldsymbol{\xi}$ also affects the distance measurement.

3.2 Shape Space Estimation by SOM^2

The SOM is an algorithm that approximates the given dataset by a finite nonlinear manifold. It is also regarded as an algorithm that estimates the map from the latent space to the observed data space. In our framework, what we want to do is estimate the homotopy from the latent product space to the observed data space. This also means that the given data distribution is approximated by a fiber bundle. The higher-rank of SOM, i.e., SOM^n, provides the exact algorithm for this purpose. In the case of a SOM^2, it consists of a set of SOMs at the 1st level (1st SOMs) and one SOM at the 2nd level (2nd SOM). The purpose of the 1st SOMs is to model each given dataset, while the 2nd SOM organizes a map of the models organized by the 1st SOMs. For the shape space estimation, the task of the 1st SOMs is to obtain the shape models, while the task of the 2nd SOM is to organize their map. In addition, the observation transformation should also be estimated in our framework. Therefore, we create an observation invariant SOM^2 algorithm by introducing the distance measure described above.

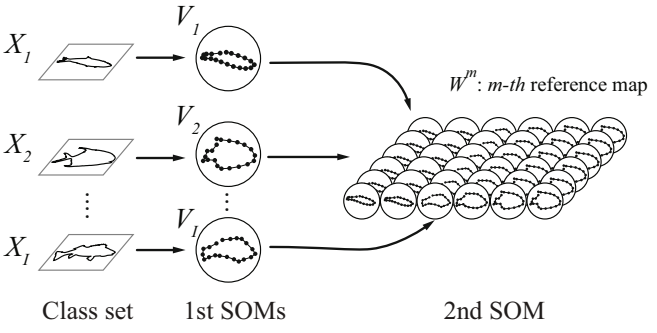


Fig. 3. Architecture of SOM^2

3.3 Observation Invariant Algorithm for SOM^2

Architecture. Now suppose that we have I observed shape data $\{X_1, \ldots, X_I\}$, each of which consists of the contour pixels, $X_i = \{\mathbf{x}_{ij}\}$, $\mathbf{x}_{ij} \in \mathbb{R}^D$. The first task of the SOM^2 is to estimate the observed shape model $\{\tilde{\mathbf{Y}}_i\}$. For this purpose, the SOM^2 has a set of I 1st SOMs, each of which consists of N units (Fig.3). Then the shape model organized by the i-th 1st SOM is represented by $\mathbf{V}_i = \bigoplus_{n=1}^{N} \mathbf{v}_i^n$. Here $\mathbf{v}_i^n$ is the reference vector of the n-th unit in the i-th 1st SOM, and the symbol $\oplus$ means the concatenation of vectors. Thus, the i-th shape is represented by the vector $\mathbf{V}_i \in \mathbb{R}^{D\times N}$.

The second task of the SOM^2 is to organize a map of $\{\mathbf{V}_i\}$. For this purpose, the 2nd SOM consists of M units, the reference vectors of which are $\mathbf{W}^m \in \mathbb{R}^{D\times N}$. Each unit of the 2nd SOM, therefore, also represents a shape. More precisely, $\mathbf{W}^n$ of the 2nd SOM is expected to represent the original shape *before* observation transformation, whereas $\mathbf{V}_i$ of the 1st SOM represents the shape *after* transformation.

Using this architecture, the algorithm for estimating the shape space is described as follows.

Step 1: Modeling the Observed Shapes. In the first step of the algorithm each observed shape is modeled by the corresponding 1st SOM. This step is described by the ordinary SOM algorithm. Thus, the winning unit is determined for each $\mathbf{x}_{ij}$, then the neighborhood function is evaluated, and finally the reference vector of each unit is updated as follows.

$$\text{winner}(\mathbf{x}_{ij}) = \arg\min_n \left\|\mathbf{x}_{ij} - \mathbf{v}_i^n\right\|^2 \tag{5}$$

$$\beta_{ij}^n = \exp\left[-\frac{1}{2\sigma_1^2}\left\|\boldsymbol{\xi}^{\text{winner}(\mathbf{x}_{ij})} - \boldsymbol{\xi}^n\right\|^2\right] \tag{6}$$

$$\mathbf{v}_i^n := \frac{\sum_{j=1}^{J_i} \beta_{ij}^n \mathbf{x}_{ij}}{\sum_{j'=1}^{J_i} \beta_{ij'}^n} \tag{7}$$

Here, $\boldsymbol{\xi}^n$ is the position of the n-th unit in the latent space of the 1st SOM. Then the observed shape model $\mathbf{V}_i$ is obtained as the concatenation of the reference vector $\{\mathbf{v}_i^n\}$.

Step 2: Estimating the Shape Space At the second level, the obtained models $\{\mathbf{V}_i\}$ are regarded as a set of data vectors, and the 2nd SOM is updated by the ordinary SOM algorithm. At first, the winning unit is determined for each shape model $\mathbf{V}_i$.

$$\text{winner}(\mathbf{V}_i) = \arg\min_m \tilde{L}\,(\mathbf{V}_i \| \mathbf{W}^m) \tag{8}$$

Here the distance measure defined by (4) is applied. The required transformation is also memorized as

$$\boldsymbol{\theta}_i = \arg\min_{\boldsymbol{\theta}} \left\| \mathbf{V}_i - P_{\boldsymbol{\theta}}\, \mathbf{W}^{\text{winner}(\mathbf{V}_i)} \right\|. \tag{9}$$

Thus the estimated observation transformation of the i-th shape becomes $P_i = P_{\boldsymbol{\theta}_i}$. Then the neighborhood function is evaluated, and every reference vector is updated.

$$\alpha_i^m = \exp\left[-\frac{1}{2\sigma_2^2} \left\| \boldsymbol{\omega}^{\text{winner}(\mathbf{V}_i)} - \boldsymbol{\omega}^m \right\|^2 \right] \tag{10}$$

$$\mathbf{W}^m := \frac{\sum_{i=1}^{I} \alpha_i^m P_i^{-1} \mathbf{V}_i}{\sum_{i'=1}^{I} \alpha_{i'}^m} \tag{11}$$

Step 3: Copy Back from the 2nd to the 1st SOM. Each observed shape represented by $\mathbf{V}_i$ is expected to be a member of the shape space expressed as $\{\mathbf{W}^m\}$. Therefore, each $\mathbf{V}_i$ is projected onto the estimated shape space with transformation P_i. This projection is described as the 'copy back process' in the SOM^2 algorithm.

$$\mathbf{V}_i := P_i\, \mathbf{W}^{\text{winner}(\mathbf{V}_i)} \tag{12}$$

This copy back process also has the effect of aligning the latent variable $\boldsymbol{\xi}$ between different shapes, so that natural fibers are organized in the fiber bundle.

Step 4: Alignment in the 2nd SOM. Since the task of the 2nd SOM is to represent the original shapes before observation transformation, it is expected that the distance defined by (4) is equal to the distance defined by (1), i.e., $L(\mathbf{W}^{m_1}, \mathbf{W}^{m_2}) = \tilde{L}(\mathbf{W}^{m_1}, \mathbf{W}^{m_2})$. Thus,

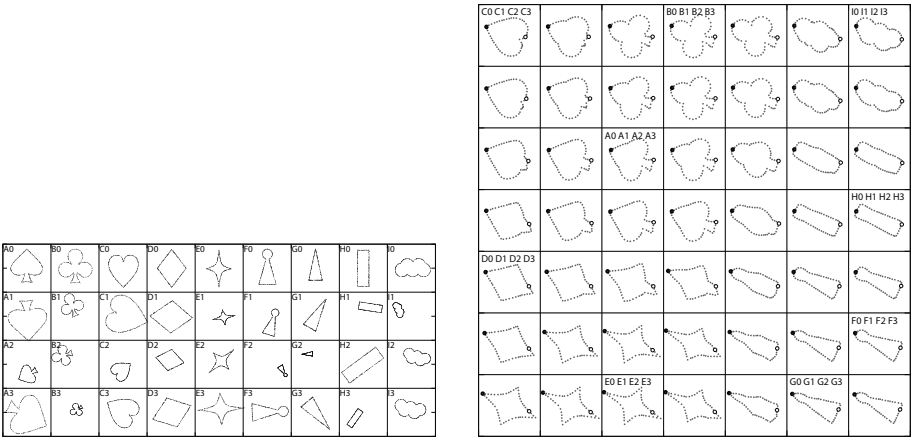
$$\|\mathbf{W}^{m_1} - \mathbf{W}^{m_2}\| = \min_{P \in \mathcal{P}} \|\mathbf{W}^{m_1} - P\, \mathbf{W}^{m_2}\| \tag{13}$$

is ideally expected. For this purpose, all reference vectors in the 2nd SOM are aligned as follows.

$$P^m = \arg\min_{P \in \mathcal{P}} \|P\, \mathbf{W}^m - \mathbf{W}^*\| \tag{14}$$

$$\mathbf{W}^m := P^m \mathbf{W}^m \tag{15}$$

Here $\mathbf{W}^*$ is the anchoring shape of the transformation. The typical anchoring shape for the contours is a unit circle. If the prototype or the representative shape is known, it can be used as the anchoring shape. For example, a typed letter 'A' could be the



(a) (b)

Fig. 4. (a) Given shapes. (b) Estimated shape space.

anchoring shape for handwritten shapes. Another candidate for the anchoring shape is the reference vector of the center unit in the 2nd SOM. By using the anchoring shape, all shape models are aligned to have almost the same scale and the same rotation angle.

The above four steps are iterated while reducing the neighborhood size until the organized map reaches a steady state.

4 Simulations and Results

4.1 Affine Invariant Shape Space Estimation

The first simulation estimates the shape space of artificial contours (Fig.4). Nine different shapes are prepared, and then transformed with respect to their positions, scales, and angles. Fig.4(a) shows the shapes used in the simulation. In this case, the observation transformation is described by a subset of affine deformation, that is,

$$P_\theta \mathbf{y} = \begin{pmatrix} a & -b & c \\ b & a & d \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ 1 \end{pmatrix}. \tag{16}$$

The parameter vector $\boldsymbol{\theta} = (a, b, c, d)$ is estimated by the Newton method.

The results are shown in Fig.4(b). In the organized 2nd map, contours are represented continuously so that a contour shape gradually morphs from one to the other. Furthermore the same shapes with different observation transformations are mapped to the same location in the shape space in the 2nd map. Thus the affine robust shape map can be successfully organized in the 2nd SOM.

4.2 Skyline Shape Map of Omnidirectional Images

In the second simulation, a set of images from an omnidirectional camera mounted on an autonomous mobile robot is used. Omnidirectional cameras are often used in mobile robots, but the image is deformed nonlinearly when the robot moves over irregular ground (Fig.5(a)). Thus the image depends on both the robot's pose and the ground condition. A set of images are generated by simulating a mobile robot's movement, and then the skyline contours are extracted. The trajectory of the robot in the field is shown in Fig.5(b).

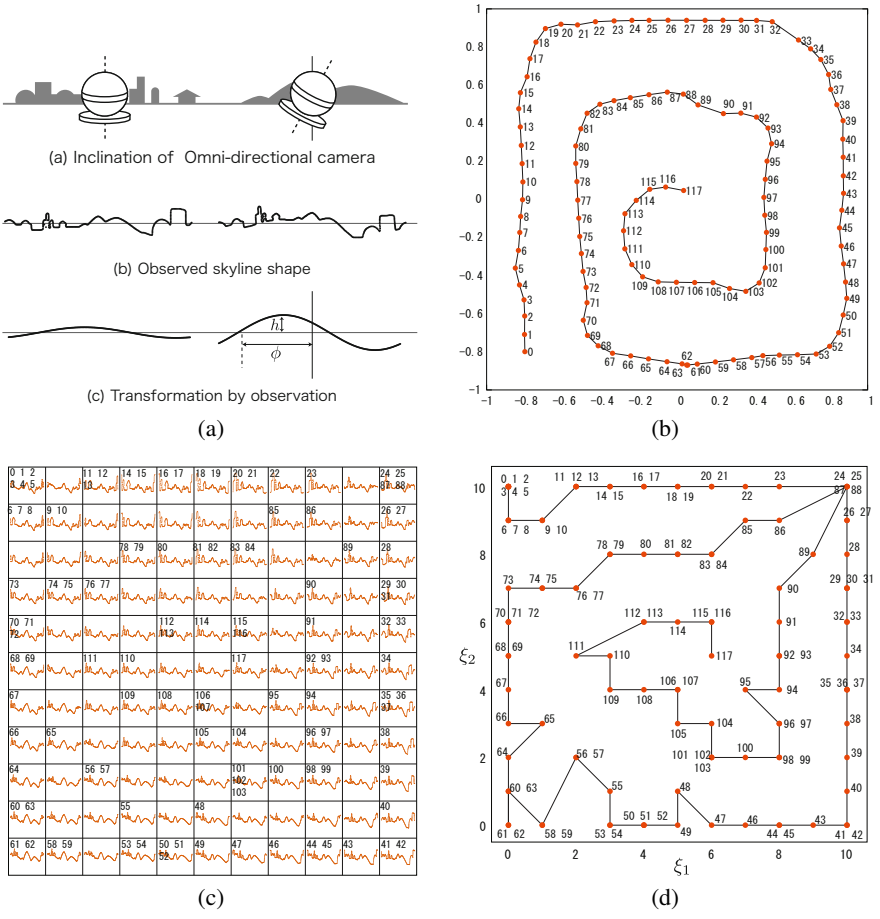


Fig. 5. Shape space of the skyline contours of an omnidirectional camera mounted on a mobile robot. (a) The omnidirectional image is nonlinearly deformed when the ground condition is irregular. (b) The trajectory of the mobile robot in the field. (c) The estimated skyline shape space. (d) The trajectory of the skyline contours in the shape space.

The estimated skyline shape space is shown in Fig.5(c). When the robot moves as shown in Fig.5(b), the skyline shape moves in the shape space as shown in Fig.5(d). The actual trajectory of the robot is roughly duplicated in the shape space. Thus the robot can localize its position in the field from the skyline shape, even if the ground condition is irregular. In addition, it is also possible to estimate the robot's pose by estimating the observation transformation.

5 Conclusion

In this paper, the concept of shape space was proposed with the generative model for shapes. As an implementation of the shape space estimation, a higher-rank of SOM, i.e., a SOM^2 was employed. The SOM^2 is not the only solution; other types of manifold learning and topology preserving mapping would also be suitable. Experiments with more realistic datasets and theoretical establishment with a fully Bayesian approach are left for future work.

This concept is not limited to visual shapes, but can be applied in the case of abstract shapes. For example, temporal trajectories in the state space can be dealt with by regarding them as shapes. Thus, this concept could be expanded to the case of dynamics.

Acknowledgement. This work is partially supported by KAKENHI 23500280 and KAKENHI 22240022.

References

1. Lin, Z., Davis, L.S.: Shape-Based Human Detection and Segmentation via Hierarchical Part-Template Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(4), 604–618 (2010)
2. Mahoor, M.H., Abdel-Mottaleb, M.: Classification and numbering of teeth in dental bitewing images. Pattern Recognition 38(4), 577–586 (2005)
3. Wei, C.H., Li, Y., Chau, W.Y., Li, C.T.: Trademark Image Retrieval Using Synthetic Features for Describing Global Shape and Interior Structure. Pattern Recognition 42(3), 386–394 (2009)
4. Macrini, D., Dickinson, S., Fleet, D., Siddiqi, K.: Bone graphs: Medial shape parsing and abstraction. Computer Vision and Image Understanding 115(7), 1044–1061 (2011)
5. Loncaric, S.: A Survey of Shape Analysis Techniques. Pattern Recognition 31(8), 983–1001 (1998)
6. Zhang, D., Lu, G.: Review of shape representation and description techniques. Pattern Recognition (37), 1–19 (2004)
7. Kwok, J.T., Tsang, I.W.: The pre-image problem in kernel methods. IEEE Transactions on Neural Networks 15(6), 1517–1525 (2004)
8. Datta, A., Pal, T., Parui, S.K.: A Modified self-organizing neural net for shape extraction. Neurocomputing (14), 3–14 (1997)
9. Kumar, G.S., Kalra, P.K., Dhande, S.G.: Curve and surface reconstruction from points: an approach based on self-organizing maps. Applied Soft Computing (5), 55–66 (2004)
10. Furukawa, T.: SOM of SOMs. Neural Networks 22(4), 463–478 (2009)

Neocognitron Trained by Winner-Kill-Loser with Triple Threshold

Kunihiko Fukushima[1,2], Isao Hayashi[2], and Jasmin Léveillé[3]

[1] Fuzzy Logic Systems Institute, Iizuka, Fukuoka 820–0067, Japan
fukushima@m.ieice.org
http://www4.ocn.ne.jp/~fuku_k/index-e.html
[2] Faculty of Informatics, Kansai University, Takatsuki, Osaka 569–1095, Japan
[3] Department of Cognitive and Neural Systems and Center of Excellence for Learning in Education, Science, and Technology, Boston University, Boston, MA 02215, USA

Abstract. The *neocognitron* is a hierarchical, multi-layered neural network capable of robust visual pattern recognition. The neocognitron acquires the ability to recognize visual patterns through learning. The winner-kill-loser is a recently introduced competitive learning rule that has been shown to improve the neocognitron's performance in character recognition. This paper proposes an improved winner-kill-loser rule, in which we use a triple threshold, instead of the dual threshold used as part of the conventional winner-kill-loser. It is shown theoretically, and also by computer simulation, that the use of a triple threshold makes the learning process more stable. In particular, a high recognition rate can be obtained with a smaller network.

Keywords: Visual pattern recognition, Neocognitron, Hierarchical network, Winner-kill-loser, Triple threshold.

1 Introduction

The *neocognitron*, originally proposed by Fukushima [1], is a hierarchical, multi-layered neural network capable of robust visual pattern recognition. The neocognitron acquires the ability to recognize patterns through learning.

During learning, input connections to feature-extracting cells are modified upon presentation of training patterns. Several methods for training the neocognitron have been proposed to date. One of them, the *winner-kill-loser* learning rule, is known to be very powerful at training intermediate stages of the hierarchical network [2].

This paper proposes an improved learning rule, in which we use a triple threshold, instead of the dual threshold associated with the original winner-kill-loser. We show theoretically, and also by computer simulation, that the use of a triple threshold makes the learning process more stable. In particular, a high recognition rate can be obtained with a smaller scale of the network.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 628–637, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Outline of the Network

The neocognitron consists of layers of S-cells, which resemble simple cells in the visual cortex, and layers of C-cells, which resemble complex cells. These layers of S-cells and C-cells are arranged alternately in a hierarchical manner.

The neocognitron discussed in this paper consists of four stages of S- and C-cell layers: $U_0 \rightarrow U_{S1} \rightarrow U_{C1} \rightarrow U_{S2} \rightarrow U_{C2} \rightarrow U_{S3} \rightarrow U_{C3} \rightarrow U_{S4} \rightarrow U_{C4}$. Here we use notation like U_{Sl}, for example, to indicate the layer of S-cells of the lth stage.

Each layer of the network is divided into a number of sub-layers, called *cell-planes*, depending on the feature to which cells respond preferentially. Incidentally, a cell-plane is a group of cells that are arranged retinotopically and share the same set of input connections [1]. As a result, all cells in a cell-plane have identical receptive fields but at different locations.

Stimulus patterns are presented to the input layer, U_0. The output of U_0 is then sent directly to U_{S1}. An S-cell in this layer resembles a simple cell in the primary visual cortex, and responds selectively to an edge at a particular orientation. To be more specific, U_{S1} has $K_{S1} = 16$ cell-planes, whose preferred orientations are chosen at an interval of 22.5°. As a result, contours in the input image are decomposed into edges for every orientation in U_{S1}. Unlike the S-cells in subsequent layers, S-cells in U_{S1} are made of analog threshold elements. Mathematically, an S-cell in U_{S1} extracts an oriented edge directly from U_0 using a linear filter followed by a half-wave rectification. There is also a mechanism of week lateral inhibition among S-cells of different preferred orientations. The shape of the linear filter is encoded in the input connections to an S-cell and is implemented as a directional derivative of two-dimensional Gaussian. The neocognitron used here thus differs from previous versions [3] in which the S-cells were the same across all layers and where an additional contrast-extracting layer, U_G, was present between U_0 and U_{S1}.

S-cells at later stages (U_{S2} to U_{S4}) are each accompanied by an inhibitory V-cell. The V-cell, whose output is proportional to the root-mean-square of its input signals, inhibits the S-cell. In the conventional neocognitron, the V-cell performed a divisive normalization operation. In the current model, however, the V-cell inhibits the S-cell in a subtractive manner, which has been shown to increase robustness to background noise [4].

At each stage of the hierarchical network, the output of layer U_{Sl} is fed into layer U_{Cl}. C-cells have fixed input connections. By averaging their input signals, C-cells exhibit some level of translation invariance. As a result of averaging across position, C-cells encode a blurred version of their input. The blurring operation is essential for endowing the neocognitron with an ability to recognize patterns robustly, with little effect from deformation, change in size, or shift in the position of input patterns. Unlike previous versions of the neocognitron that used the arithmetic mean, in the current model averaging is done through root-mean-square [4].

As in previous versions of the neocognitron, excitatory connections to the C-cells in U_{C1} and U_{C2} are surrounded by inhibitory connections, yielding concentric on-center off-surround connections.

The strength of input connections to S-cells are modified through learning. After learning, S-cells become feature-extracting cells. S-cells at higher stages of the hierarchy extract more global features than S-cells at lower stages.

Learning is performed layerwise, from lower layers to higher layers, such that the training of a given stage can start only after the training of the preceding stage is complete. In order to train S-cells in U_{Sl}, the responses of C-cells in the preceding layer U_{Cl-1} are used as a training stimulus. All S-cell layers, except for U_{S1}, are trained with the same training set.

We use the *winner-kill-loser* rule, a variant of competitive learning, to train intermediate layers U_{S2} and U_{S3} [2]. We propose to use a triple threshold to govern learning, instead of the dual threshold that was used in the original winner-kill-loser rule. This is the main topic of this paper and is discussed below in more detail.

As mentioned above, each layer of the neocognitron is divided into cell-planes. All cells in a cell-plane share the same set of input connections. This condition of shared connections has to be kept even during the learning phase, when input connections to S-cells are renewed. When a winner is chosen in a given cell-plane, its input connections are renewed based on the responses of the C-cells presynaptic to it. Since all cells in the cell-plane share the same set of connections, all other cells in the cell-plane come to have the same connections as the winner. The winner thus works like a seed in crystal growth. Hence we call it a *seed-cell*.

S-cells at the highest stage (U_{S4}) are trained by supervised competitive learning using labeled training data [3]. As the network learns varieties of deformed training patterns, more than one cell-plane per class is usually generated in U_{S4}.

Every time a training pattern is presented, competition occurs among all S-cells in the layer. If the winner of the competition has the same label as the training pattern, the winner becomes the seed-cell and learns the training pattern. However, if the winner has a wrong label (or if all S-cells are silent), a new cell-plane is generated. The new cell-plane hence learns the current training pattern simply by being assigned its corresponding label. Each cell-plane of U_{S4} thus has a label indicating one of the 10 digits.

During the recognition phase, the label of the maximally activated S-cell in U_{S4} determines the final result of recognition. The C-cells at the highest stage yield the inferred label of the input stimulus.

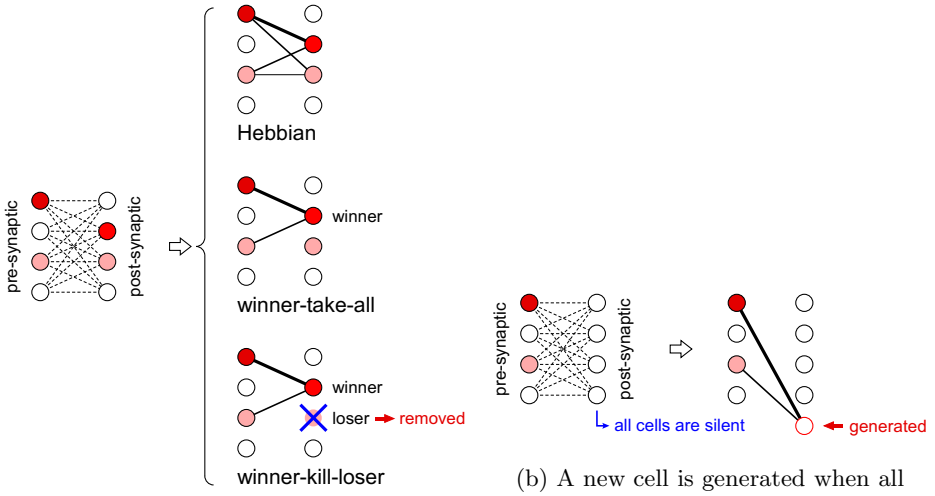
3 Competitive Learning with Winner-Kill-Loser

3.1 Winner-Kill-Loser with Dual Threshold

In order to train S-cells in layers U_{S2} and U_{S3}, we use competitive learning with winner-kill-loser [2]. We first explain the original winner-kill-loser rule, which uses a dual threshold for S-cells.

Fig. 1 illustrates the learning process with the original winner-kill-loser rule [2], and compares it with other learning rules.

The Hebbian rule, shown at the top of Fig. 1(a), is one of the most commonly used learning rules. During the learning phase, each synaptic connection



(a) Several rules of learning

(b) A new cell is generated when all postsynaptic cells are silent

Fig. 1. Winner-kill-loser rule with dual threshold, in comparison with other learning rules. In this figure, the response of each cell is represented by the saturation of the color.

is strengthened by an amount proportional to the product of the responses of the pre- and post-synaptic cells.

In the winner-take-all rule, shown in the middle of Fig. 1(a), post-synaptic cells compete with each other, and the cell from which the largest response is elicited becomes the winner. Only the winner can have its input connections renewed. The magnitude of the weight change is proportional to the response of the pre-synaptic cell. Incidentally, most of the conventional neocognitrons [1,3] use this learning rule.

The winner-kill-loser rule, shown at the bottom of Fig. 1(a), resembles the winner-take-all rule in the sense that only the winner learns the training stimulus. In the winner-kill-loser rule, however, not only does the winner learn the training stimulus, but also the losers are simultaneously removed from the network. Losers are defined as cells whose responses to the training stimulus are smaller than that of the winner, but whose activations are nevertheless greater than zero.

If a training stimulus elicits non-zero responses from two or more S-cells, it means that the preferred features of these cells resemble each other, and that they work redundantly in the network. To reduce this redundancy, only the winner has its input connections renewed to fit more to the training vector, while the other active cells, namely the losers, are removed from the network.

Since silent S-cells (namely, the S-cells whose responses to the training stimulus are zero) do not join the competition, they are not removed. These cells are expected to work toward extracting other features.

As depicted in Fig. 1(b), a new S-cell is generated if all cells are silent for a given training stimulus. The initial value of the input connections of the newly generated S-cell is proportional to the response of the pre-synaptic cells.

Incidentally, the generation of new S-cells was also a feature of the winner-take-all rule implemented in previous versions of the neocognitron.

In the learning phase, a number of training stimuli are presented sequentially to the network. During this process, generation of new cells and removal of redundant cells occurs repeatedly in the network. In particular, new cells are generated to cover areas of the multi-dimensional feature space that were not previously covered by existing cells. In the areas where similar cells exist in duplicate, redundant cells are removed. By repeating this process for a long enough time, the preferred features (reference vectors) of S-cells gradually become distributed uniformly over the multi-dimensional feature space.

When applying this learning rule to the neocognitron, a slight modification is required due to the fact that each layer of the network consists of a number of cell-planes such that all cells in a given cell-plane must share the same set of input connections both during learning and recognition.

At first, the S-cell whose response is the largest in the layer is chosen as a seed-cell. The seed-cell has its input connections renewed depending on the training vector presented to it. Once the connections to the seed-cell are renewed, all cells in the cell-plane from which the seed-cell is chosen come to have the same set of input connections as the seed-cell because of the shared connections. All non-silent cells at the same spatial location as the seed cell are determined losers, and the cell-planes to which they belong are removed from the layer.

In the original winner-kill-loser rule, a dual threshold is used to guide learning and recognition in S-cells [2]. Namely, S-cells have a higher threshold during learning (θ^{L}) than during recognition (θ^{R}). During learning, S-cells join the competition only when their responses to a training vector are not zero under the high threshold θ^{L}. This means that, even though an S-cell would yield a non-zero response under the low recognition threshold θ^{R}, that S-cell does not join the competition provided it is silent under the high learning threshold θ^{L}.

It has been demonstrated by computer simulation that the use of the winner-kill-loser rule in the competitive learning largely increases the recognition rate for smaller network sizes [2]. However, there still remain some problems in the winner-kill-loser with dual threshold.

Problems with the Dual Threshold Formulation. As mentioned above, the original winner-kill-loser makes use of a dual threshold: a high threshold θ^{L} for learning and a low threshold θ^{R} for recognition. That is, in the learning phase, only one threshold, θ^{L}, is used.

We now use vector notation. Let $\boldsymbol{x}$ be the input signal from a set of presynaptic C-cells to an S-cell. We use vector $\boldsymbol{X}$ to represent the strength of input connections of the S-cell. We can interpret $\boldsymbol{X}$ as the preferred (optimal) feature of the S-cell in a multi-dimensional feature space. We sometimes call $\boldsymbol{X}$ the *reference vector* of the S-cell.

When a training vector $\boldsymbol{x}$ is presented, an S-cell calculates the similarity s between $\boldsymbol{X}$ and $\boldsymbol{x}$ by

$$s = (\boldsymbol{X}, \boldsymbol{x}) \,/\, \{\|\boldsymbol{X}\| \cdot \|\boldsymbol{x}\|\} \,. \tag{1}$$

If similarity s is larger than θ^{L}, the S-cell with subtractive inhibition yields a non-zero response [4]

$$u = \|\boldsymbol{x}\| \cdot \frac{\varphi[s - \theta^{\mathrm{L}}]}{1 - \theta^{\mathrm{L}}} \ , \tag{2}$$

where $\varphi[\ ]$ is a function defined as $\varphi[x] = \max(x, 0)$. The area that satisfies $s > \theta^{\mathrm{L}}$ in the multi-dimensional feature space is called the *tolerance area* of the S-cell. This situation is illustrated in Fig. 2.

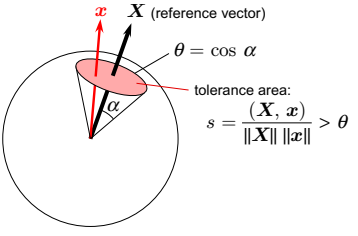


Fig. 2. Tolerance area of an S-cell in the multi-dimensional feature space

If the response of the S-cell is the largest among all S-cells, the S-cell learns the training vector by adding $\boldsymbol{x}$ to $\boldsymbol{X}$. Other S-cells with non-zero response are determined losers and are removed from the network. If all S-cells are silent, a new cell is generated, and $\boldsymbol{x}$ becomes the reference vector of the generated S-cell.

It is expected that the learning process produces a situation where reference vectors of S-cells distribute uniformly in the multi-dimensional feature space as shown in Fig. 3(a).

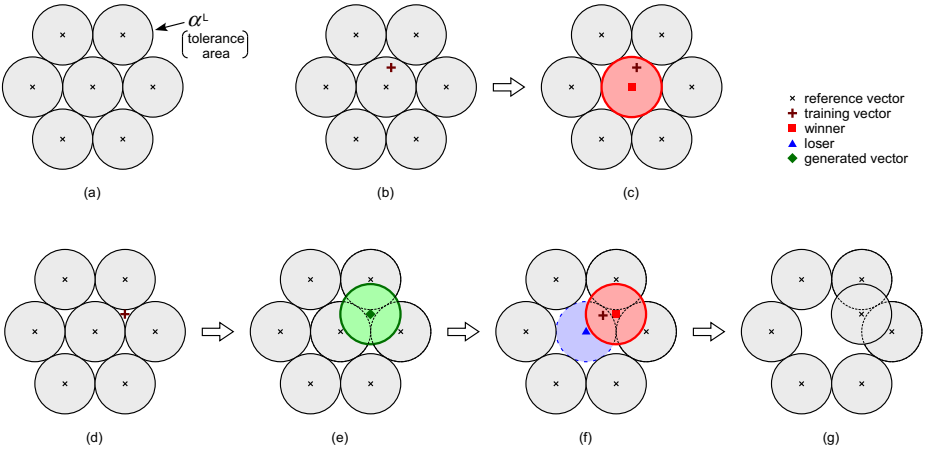


Fig. 3. Progress of learning by winner-kill-loser with a dual threshold. The dual threshold means that, in the learning phase, only one threshold, θ^{L}, is used.

Let us now observe how the distribution of reference vectors changes during learning. We assume that, at a certain moment in the learning, we happen to have a uniform distribution of reference vectors as shown in Fig. 3(a). If a training vector is presented at ✚ in (b) of the figure, the cell at ■ is the winner and learns the training vector as shown in (c). This is all right.

If a training vector is presented at ✚ as shown in (d), however, all cells are silent and a new cell, whose reference vector is at ◆, is generated as shown in (e). After that, if another training vector is presented at ✚ in (f), the cell at ■ becomes a winner and the cell at ▲ becomes a loser. Removal of the loser results in the situation depicted in (g). Thus, further training can actually destroy the desirable uniform distribution that was present in (a).

This means that removal and generation of cell-planes in the neocognitron do not stabilize during learning. Since the number of cell-planes continues to increase and decrease, the final number of cell-planes obtained after learning is determined largely by the duration of the learning phase. This in turn will strongly affect the network's recognition rate and scale.

3.2 Use of Triple Threshold for Winner-Kill-Loser

We now propose to add one more threshold during learning. In other words, we use thresholds θ^{W} and θ^{G}, instead of only one threshold θ^{L} (Fig. 4).

	dual threshold	triple threshold
learning phase	θ^{L}	θ^{W} for choosing a winner & losers
		θ^{G} for generating a new cell-plane
recognition phase	θ^{R}	θ^{R}

Fig. 4. Comparison of dual and triple thresholds

Threshold θ^{W} is used for determining the winner and losers. Non-zero responses are elicited from S-cells whose similarity s (namely, similarity between reference vector $\boldsymbol{X}$ of the cell and the training vector $\boldsymbol{x}$) is larger than θ^{W}. Among these S-cells, an S-cell that has the largest response becomes the winner and learns $\boldsymbol{X}$. Other non-silent S-cells are categorized as losers and are removed from the network.

The threshold θ^{G} works like a kind of subliminal threshold and controls the generation of a new S-cell (namely, the generation of a new reference vector in the vector space). If there exists at least one S-cell whose similarity s is larger than θ^{G}, no new S-cell is generated. In other words, not only an active S-cell that has $s > \theta^{\mathrm{W}}$, but also a silent S-cell whose similarity s is in the range $\theta^{\mathrm{W}} \geq s > \theta^{\mathrm{G}}$, can prevent the generation of a new S-cell. A new S-cell can be generated only when the similarity s of all S-cells are under the threshold θ^{G}. This situation is illustrated in Fig. 5.

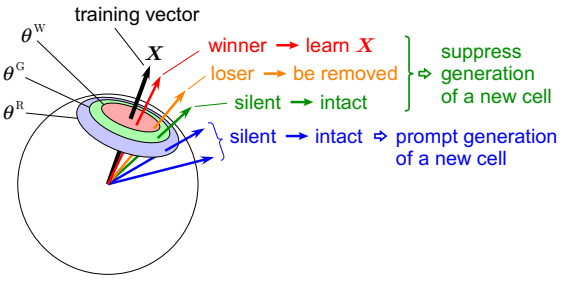


Fig. 5. Winner-kill-loser with triple threshold in the multi-dimensional vector space. Here we propose to use a subliminal threshold, θ^G, such that the presence of a cell whose similarity is greater than θ^G prevents the generation of a new cell.

Optimal Values of the Thresholds. We now discuss how to choose threshold values θ^W and θ^G. To represent the radius of tolerance area, which is determined by threshold θ, we use angle α between two vectors in the multi-dimensional feature space, as shown in Fig. 2. Namely, $\theta^W = \cos\alpha^W$, and $\theta^G = \cos\alpha^G$. Although the feature space is actually of dimensionality greater than two, for simplicity we start our discussion assuming that it is a two-dimensional plane.

The goal of training is to make reference vectors distribute uniformly in the feature space. Once a desirable uniform distribution of reference vectors has emerged during learning, it should not be destroyed upon further training. To prevent reference vectors from becoming losers and being removed, the tolerance areas of radius α^W should not overlap (Fig. 6). In a feature space of dimensionality greater than one, however, it is inevitable that some vacant gaps are generated between non-overlapping disks of radius α^W. To prevent generation of a new reference vector, the feature space is covered by disks of radius α^G that can overlap with each other. The smallest α^G that can fill vacant gaps can be determined from α^W, as illustrated in the right of Fig. 6. Namely,

$$\alpha^W = \alpha^G \cos(\pi/6)\,, \qquad \text{or} \qquad \cos^{-1}\theta^W = \cos(\pi/6)\cos^{-1}\theta^G\,. \tag{3}$$

Simulations were conducted to assess the impact of the dual and triple thresholds in the neocognitron trained with the winner-kill-loser rule. The training set we use consists of 3000 handwritten digits (300 patterns for each digit) randomly sampled from the ETL1 database [5]. This training set is presented only once to train layers U_{S2} and U_{S3}. Fig. 7 shows how the number of cell-planes in layer U_{S3} changes during learning. Results for the triple and dual thresholds are depicted as a red solid line and a blue dotted line, respectively. It can be seen from the figure that the fluctuation of the number of cell-planes is much smaller with the triple threshold and that the learning can progress more stably. The final number of cell-planes that has been created after learning is usually smaller with the triple threshold than with the dual threshold.

Fig. 8 shows how the error rate of the neocognitron changes with the size of the training set. The test set consists of 5000 patterns (500 patterns for each digit). Experiments were repeated twice for each condition, using different learning and

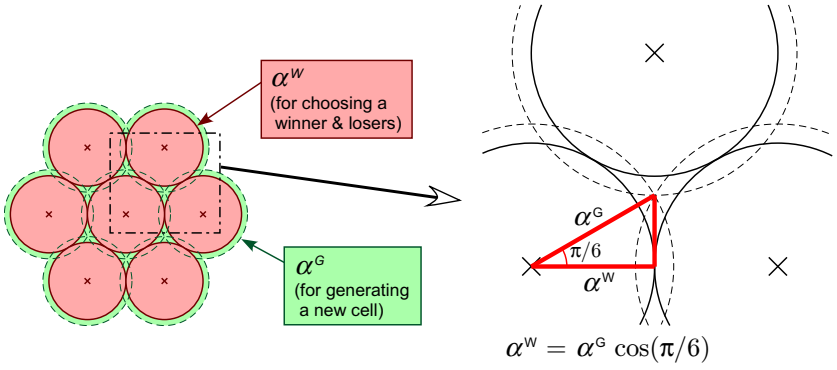


Fig. 6. Winner-kill-loser with triple threshold. Use of thresholds $\theta^{\mathrm{G}}(= \cos\alpha^{\mathrm{G}})$ and $\theta^{\mathrm{W}}(= \cos\alpha^{\mathrm{W}})$ for the learning.

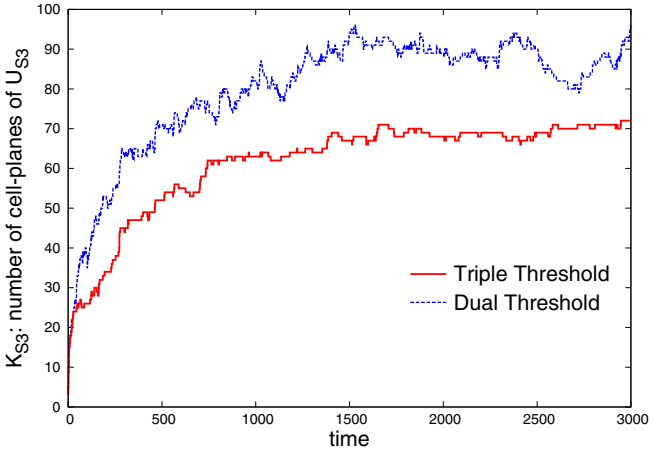


Fig. 7. Number of cell-planes (K_{S3}) during the learning

test sets randomly sampled from the ETL1 database [5], and the results were averaged across the two experiments. We can see that the recognition error itself does not differ much whether using the dual (blue dotted line) or triple threshold (red solid line). Although the recognition error of the neocognitron depends on the final number of cell-planes that have been created after learning, we usually have almost the same recognition rate with a smaller number of cell-planes when the network is trained with the triple threshold.

In the case of the result shown in Fig. 7, for example, when using the triple threshold, the final number of cell-planes in each layer was $(K_{\mathrm{S2}}, K_{\mathrm{S3}}, K_{\mathrm{S4}}) = (31, 72, 73)$, and the recognition error was 1.22%. On the other hand, under the dual threshold, we had $(K_{\mathrm{S2}}, K_{\mathrm{S3}}, K_{\mathrm{S4}}) = (30, 96, 82)$ and 1.26%, respectively. It should be noted here that the computational cost for calculating the response of $\mathrm{U}_{\mathrm{S}l}$ is approximately proportional to $K_{\mathrm{S}l-1} \times K_{\mathrm{S}l}$.

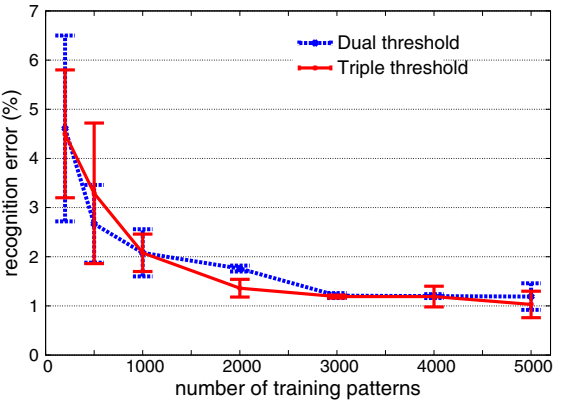


Fig. 8. Recognition error vs. number of training patterns

4 Discussions

In this paper we introduce a new triple threshold to be used for competitive learning with the winner-kill-loser rule. We show by computer simulation that the use of the triple threshold makes the learning process more stable than when using the dual threshold. Although the triple threshold does not seem to improve recognition rate, it nevertheless significantly reduces network scale (with a smaller number of cell-planes in each layer). One of the greatest merits of the triple threshold formulation is the stability of learning, which in turn makes the neocognitron less sensitive to the duration of the learning phase.

Acknowledgements. This work was partially supported from Kansai University by Strategic Project to Support the Formation of Research Bases at Private Universities: Matching Fund Subsidy from MEXT, 2008–2012.

References

1. Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 36(4), 193–202 (1980)
2. Fukushima, K.: Neocognitron trained with winner-kill-loser rule. Neural Networks 23(7), 926–938 (2010)
3. Fukushima, K.: Neocognitron for handwritten digit recognition. Neurocomputing 51, 161–180 (2003)
4. Fukushima, K.: Increasing robustness against background noise: visual pattern recognition by a neocognitron. Neural Networks 24(7), 767–778 (2011)
5. ETL1 database, http://www.is.aist.go.jp/etlcdb/#English

Nonlinear Nearest Subspace Classifier

Li Zhang[1], Wei-Da Zhou[2], and Bing Liu[3]

[1] Research Center of Machine Learning and Data Analysis, School of Computer Science and Technology, Soochow University, Suzhou 215006, Jiangsu, China
[2] AI Speech Ltd., Suzhou 215123, Jiangsu, China
[3] Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China

Abstract. As an effective nonparametric classifier, nearest subspace (NS) classifier exhibits its good performance on high-dimensionality data. However, NS could not well classify the data with the same direction distribution. To deal with this problem, this paper proposes a nonlinear extension of NS, or nonlinear nearest subspace classifier. Firstly, the data in the original sample space are mapped into a kernel empirical mapping space by using a kernel empirical mapping function. In this kernel empirical mapping space, NS is then performed on these mapped data. Experimental results on the toy and face data show this nonlinear nearest subspace classifier is a promising nonparametric classifier.

Keywords: Nearest subspace classifier, Kernel empirical mapping, Least square method, Machine learning.

1 Introduction

At present, subspace learning has attracted substantial attention in machine learning and computer vision. The idea of nearest subspace (NS) classification is very popular [9],[10],[3],[8]. Here, subspaces mean not only the (reduced) low-dimensional space but also the space spanned by samples per class. In [10], the nearest feature line (NFL) algorithm is presented to classify a test sample by finding the minimum distance between the test sample and any pair of training samples belonging to the same class. The nearest feature plane (NFP) and nearest feature space (NFS) classifiers are proposed as extensions of NFL in [3]. In these two methods, the distance is computed between a test sample and any plane or space spanned by the training samples per class, respectively. In addition, NFS uses at least four feature training samples to span subspaces in each class. The subspaces are typically spanned by five to nine training samples in [8], and spanned by all the samples per class in [9]. Here, we consider the NS classifier used in [9],[12],[18],[4], i.e., the subspace spanned by all training samples per class. The NS classifier is to classify a test sample based on the best linear representation in terms of all the training samples in each class [12]. There is no training procedure, so NS is a nonparametric learning method, like nearest neighbor (NN) [6] and sparse representation-based classifier (SRC) [12]. The test sample is classified based on the best representation in terms of a single training sample in NN, and based on the best sparse representation according to all training samples of all classes in SRC. NS has been widely applied to face recognition [9],[3],[12],[8], microarray cancer datasets [4],

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 638–645, 2011.
© Springer-Verlag Berlin Heidelberg 2011

and credit risk evaluation [18]. However, NS loses it classification ability even for the linearly separable tasks in which the data from different classes have the same direction.

This paper deals with this problem occurred in NS by introducing the kernel empirical mapping, and proposes a nonlinear extension of NS classifier, or nonlinear NS (NNS) classifier. In this method, we firstly map data in the original sample space into a kernel empirical mapping space by using some kernel empirical mapping function. Next, the NS classifier is adopted to classify these mapped samples.

2 Nonlinear Nearest Subspace Classifier

This section proposes a nonlinear extension of NS classifier, NNS classifier. Firstly, the kernel trick and kernel empirical mapping is briefly reviewed in this section. Then we discuss the construction of NNS classifier.

2.1 Kernel Empirical Mapping

The kernel trick is a very popular technique in machine learning and pattern recognition. Typically, the role of the kernel trick is to generalize a linear algorithm to a nonlinear one. It has been successful applied to SVM, KPCA and KFDA. In these kernel methods, only Mercer kernels, kernels satisfying Mercer's condition, are used. In other words, a Mercer kernel is continuous, symmetric, positive semi-definite kernel function. Usually, a Mercer kernel can be expressed as

$$k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^T \Phi(\mathbf{x}') \tag{1}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are any two points in $\mathcal{X}$, Φ is a nonlinear mapping and $\Phi(\mathbf{x})$ is the image of $\mathbf{x}$ in the feature space. Some common used nonlinear kernels include polynomial kernels, Gaussian radial basis function (RBF) kernels, and wavelet kernels [15],[16]. The polynomial kernels have the form $k(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T\mathbf{x}' + 1\right)^d$ and RBF kernels can be expressed as $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma\|\mathbf{x} - \mathbf{x}'\|_2^2\right)$, where $d \in \mathbb{N}$ and $\gamma > 0$ are the kernel parameters.

In kernel methods, we don't know what Φ is and just adopt the kernel function (1). Thus, we can not get the mapped feature space, which makes the operation on the images of samples difficult. Zhang et al. construct a family of empirical mapping with kernel functions, and use them in SVM [14], PCA [17], and FDA [13]. The kernel empirical mapping can be explicitly computed in an empirical mapping space (feature space). Typically, the kernel empirical mapping on the data set $X = \{\mathbf{x}\}_{i=1}^n$ is:

$$\Phi(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \cdots, k(\mathbf{x}_n, \mathbf{x})]^T \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^m$ is an arbitrary sample in the input space, and n is the number of samples. From (2), we can see that the dimensionality of the kernel empirical mapping feature space is n. If $m > n$, then a dimensionality reduction is performed by (2). Importantly, it doesn't require that kernel functions used in (2) must satisfy Mercer's condition [14].

2.2 Nonlinear Nearest Subspace Classifier

Assume that there are c training subsets $\{X_1, X_2, \cdots, X_c\}$, where the jth class training subset $X_j = \{\mathbf{x}_{j,i}, y_{j,i} = j\}_{i=1}^{n_j}$ is a subspace, $\mathbf{x}_{j,i} \in \mathcal{X} \subset \mathbb{R}^m$, $y_{j,i} \in \{1, \cdots, c\} \subset \mathbb{N}$ are labels corresponding to $\mathbf{x}_{j,i}$, n_j is the number of the jth class training samples, $\mathcal{X}$ is the input space, m is the dimensionality of the sample space $\mathcal{X}$ and c is the number of classes. Given a test sample $\mathbf{x} \in \mathcal{X}$, the goal is to assign the label y of $\mathbf{x}$ according to the reconstruction errors between $\mathbf{x}$ and its approximations.

By introducing kernel empirical mapping, we map the samples in the input subspaces into a feature subspace as follows.

$$\Phi : \mathbf{x}_{j,i} \in \mathcal{X}_j \rightarrow \Phi\left(\mathbf{x}_{j,i}\right) = \left[k\left(\mathbf{x}_{1,1}, \mathbf{x}_{j,i}\right), \cdots, k\left(\mathbf{x}_{1,n_1}, \mathbf{x}_{j,i}\right), \cdots, k\left(\mathbf{x}_{c,n_c}, \mathbf{x}_{j,i}\right)\right]^T \in \mathcal{F}_j \quad (3)$$

where $\Phi\left(\mathbf{x}_{j,i}\right) \in \mathbb{R}^n$ is the image of $\mathbf{x}_{j,i}$ in the feature subspace $\mathcal{F}_j$, and $n = \sum_{j=1}^{c} n_j$. Note that the mapping is performed on the whole training samples instead of the jth class samples. Actually, the kernel function $k(\mathbf{x}, \mathbf{x}')$ can measure a relationship between $\mathbf{x}$ and $\mathbf{x}'$, e.g. similarity relationship. Thus, the image $\Phi\left(\mathbf{x}_{j,i}\right)$ reflects the relationship between $\mathbf{x}_{j,i}$ and whole training samples, and can be taken as the globally nonlinear features of $\mathbf{x}_{j,i}$ to a certain extent.

Now we construct the jth sample matrix from the jth training samples in $\mathcal{F}_j$. Namely,

$$\mathbf{K}_j = \left[\Phi(\mathbf{x}_{j,1}), \Phi(\mathbf{x}_{j,2}), \cdots, \Phi(\mathbf{x}_{j,n_j})\right] \in \mathbb{R}^{n \times n_j}, j = 1, \cdots, c \quad (4)$$

where each column denotes an image corresponding to a training sample. Obviously, $\mathbf{K}_j$ is full rank in column since $n > n_j$. Now, we represent the test sample linearly by all images in a feature subspace, and then have

$$\Phi(\mathbf{x}) = \mathbf{K}_j \boldsymbol{\alpha}_j, j = 1, \cdots, c \quad (5)$$

where $\boldsymbol{\alpha}_j \in \mathbb{R}^{n_j}$ is the weight vector of the jth feature subspace, and

$$\Phi(\mathbf{x}) = \left[k\left(\mathbf{x}_{1,1}, \mathbf{x}\right), \cdots, k\left(\mathbf{x}_{1,n_1}, \mathbf{x}\right), k\left(\mathbf{x}_{2,1}, \mathbf{x}\right), \cdots, k\left(\mathbf{x}_{c,n_c}, \mathbf{x}\right)\right]^T \quad (6)$$

As mentioned above, $\mathbf{K}_j$ is not a square matrix, so we can not also directly solve the linear equation set (5). By using the least square method, we construct the objective function

$$\min_{\boldsymbol{\alpha}_j} \ \|\Phi(\mathbf{x}) - \mathbf{K}_j \boldsymbol{\alpha}_j\|_2^2, \quad j = 1, \cdots, c \quad (7)$$

Likewise, the solution to (7) can be computed as

$$\boldsymbol{\alpha}_j = \left(\mathbf{K}_j^T \mathbf{K}_j\right)^{\dagger} \mathbf{K}_j^T \Phi(\mathbf{x}) \quad (8)$$

or

$$\boldsymbol{\alpha}_j = \left(\mathbf{K}_j^T \mathbf{K}_j + \sigma \mathbf{I}\right)^{-1} \mathbf{K}_j^T \Phi(\mathbf{x}) \quad (9)$$

where $(\cdot)^{-1}$ denotes the inverse of a matrix, $\sigma \geq 0$ is a very small constant, say 10^{-8}, and $\mathbf{I}$ is the identity matrix. The approximations or reconstructions of $\mathbf{x}$ in feature subspaces

are $\mathbf{K}_j\boldsymbol{\alpha}_j$, $j = 1, \cdots, c$. Among them, the best reconstruction with minimal reconstruction error is selected. The reconstruction error (residual) is defined as the Euclidean distance from the image to its reconstruction. Namely,

$$\delta_j = \|\Phi(\mathbf{x}) - \mathbf{K}_j\boldsymbol{\alpha}_j\|_2, j = 1, \cdots, c \tag{10}$$

By using which, we assign the label of the nearest feature subspace to $\mathbf{x}$, or

$$\hat{y} = \arg \min_{j=1,\cdots,c} \delta_j \tag{11}$$

The complete classification procedure of NNS is shown in Algorithm 1. The step 5 in Algorithm 1 makes all projected samples lie on a unit hypersphere. Note that the step 4 in Algorithm 1 is an optional one. If we don't choose a projection method, then $\mathbf{P}_j$ is assigned the identity matrix. Clearly, the feature space $\mathcal{F}$ has the dimensionality of n. As stated previously, the dimensionality of the samples is already reduced for small sample size problems in which $m > n$ when applying kernel empirical mapping. Of course, n is definitely larger than n_j. If n is too large, we can perform dimensionality reduction by using any possible projection methods, such as principle component analysis (PCA) [6], Fisher discriminant analysis (FDA) [6], [1], and even random projection (RP) [12].

Algorithm 1. Nonlinear nearest subspace method

1. **Input**: A set of training sample subsets $\{X_j\}_{j=1}^c$, where the subset $X_j = \{\mathbf{x}_{j,i}, y_{j,i} = j\}_{i=1}^{n_j}$,$\mathbf{x}_{j,i} \in \mathbb{R}^m$, a test sample $\mathbf{x} \in \mathbb{R}^m$, and let $\sigma > 0$.
2. Select a Mercer kernel $k(\cdot,\cdot)$ and its parameters.
3. Map samples in the input space into a feature space. we get the images of the training samples $\mathbf{x}_{j,i}$:

$$\Phi\left(\mathbf{x}_{j,i}\right) = \left[k\left(\mathbf{x}_{1,1}, \mathbf{x}_{j,i}\right), \cdots, k\left(\mathbf{x}_{2,1}, \mathbf{x}_{j,i}\right), \cdots, k\left(\mathbf{x}_{c,n_c}, \mathbf{x}_{j,i}\right)\right]^T, i = 1, \cdots, n_j, j = 1, \cdots, c,$$

the sample matrices $\mathbf{K}_j$, $j = 1, \cdots, c$, and the image of the test sample $\mathbf{x}$

$$\Phi(\mathbf{x}) = \left[k(\mathbf{x}_{1,1}, \mathbf{x}), \cdots, k\left(\mathbf{x}_{1,n_1}, \mathbf{x}\right), k\left(\mathbf{x}_{2,1}, \mathbf{x}_{j,i}\right), \cdots, k(\mathbf{x}_{c,n_c}, \mathbf{x})\right]^T.$$

4. [Optional] Select a projection method and get the corresponding projection matrix $\mathbf{P}_j$.
5. Normalize the columns of $\mathbf{P}_j^T\mathbf{K}_j$, $j = 1, \cdots, c$ and $\mathbf{P}_j^T\Phi(\mathbf{x})$ to have unit ℓ_2-norm.
6. Solve the least square problem (7) to get the weight vectors $\boldsymbol{\alpha}_j$, $j = 1, \cdots, c$ according to (9) or (8).
7. Compute the c reconstruction errors $\delta_j = \|\mathbf{P}_j^T\Phi(\mathbf{x}) - \mathbf{P}_j^T\mathbf{K}_j\boldsymbol{\alpha}_j\|_2$, $j = 1, \cdots, c$.
8. **Output**: The estimated label $\hat{y}$ for $\mathbf{x}$ according to (11).

3 Numerical Experiments

To validate the proposed NNS classifier, we perform numerical experiments on toy and face data sets, and compare NNS with other nonparametrical methods. All numerical experiments are performed on the personal computer with a 1.8GHz Pentium III and

1G bytes of memory. This computer runs on Windows XP, with MATLAB 7.01 and VC++ 6.0 compiler installed.

3.1 Setting of Algorithms and Their Parameters

In our experiments, some nonparametric classifiers are compared with NNS, including SRC, NN and NS. The algorithms and their parameters are described as follows.

1. SRC [12] is a new nonparametric classifier which uses sparse signal reconstruction method to sparsely represent a test point. SRC can be formulated as a quadratically constrained ℓ_1-minimization problem, which here is solved by exploiting ℓ_1-MAGIC software package [2]. Let the parameter $\epsilon = 0.001$ in [12].
2. NN [6] is an old nonparametric method, but it is very simple and efficient. The Euclidean distance is taken as a distance measurement. The number of nearest neighbors is one.
3. NS [9] can be regarded as an extension of NN. Here, NS uses the method similar to (9) to get its solution. Let $\delta = 10^{-8}$.
4. NNS proposed here is a nonlinear extension of NS. (9) are adopted to obtain the solution of NNS. Let $\delta = 10^{-8}$. We take into account the NNS with polynomial kernel (Poly-NNS) and the NNS with RBF kernel (RBF-NNS) in our experiments. The RBF kernel parameter γ is set by the median value of $\frac{1}{\|\Phi(\mathbf{x}_i)-\overline{\Phi(\mathbf{x})}\|^2}$, $i = 1, \cdots, n$, where $\overline{\Phi(\mathbf{x})}$ is the mean of all training samples.

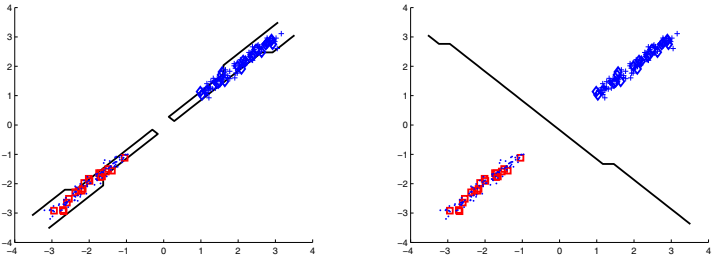
3.2 Synthetic Data Set

In the synthetic data set, there are two-class data X_1 and X_2 with m-dimensionality, where m takes value from $\{2^1, 2^2, \cdots, 2^7\}$. Each feature in X_1 and X_2 takes value from the interval $[-3, -1]$ and $[1, 3]$, respectively, and is corrupted by Gaussian noise with zero mean and 0.01 variance. We generate 20 training and 100 test points in X_1 and X_2, respectively. Figs. 1(a)-1(b) show the case of two-dimensional data X_1 and X_2, and the decision boundaries obtained by NNS and NS methods in one trial. In Fig. 1(a), all data surrounded by boundaries are classified to the same class by NS. Fig. 1(b) shows that RBF-NNS perfectly classifies this data set.

The average test errors on 100 runs are reported in Table 1. From Table 1, we can also see as the increasing of feature dimensions, the classification errors of two methods are almost unchanged. RBF-NNS has a zero error, and NS has about 50% error rate. This data set obviously is linearly separable, but NS could not solve it well.

Table 1. Mean and standard deviation of test error rate (%) on the synthetic data set

Method	Dimensionality						
	2	4	8	16	32	64	128
NS	50.54±2.55	50.24±2.96	50.25±3.18	50.19±2.84	49.96±3.14	49.28±3.14	50.06±3.93
RBF-NNS	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00



(a) Decision boundary obtained by NS (b) Decision boundary obtained by NNS

Fig. 1. Comparison of NNS with NS on the synthetic data set. The bolded lines are decision boundaries. Training data are denoted by "□" and "◊", respectively; Test data are denoted by "·" and " + ", respectively.

3.3 Face Data Sets

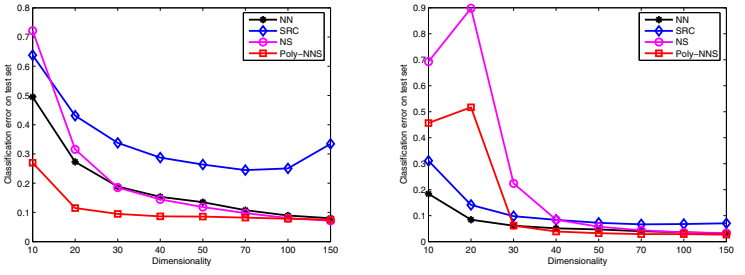
It is well known that NS have good performance on face data, so we perform experiment on two face databases, including ORL face database [11], and UMIST face database [7].Here, NNS is compared with nonparametric learning methods (NN, SRC, and NS). The original features of each face image are obtained by stacking its columns. Then for a $m_1 \times m_2$ gray-scale face image, we get $m_1 m_2$ features which are normalized by 255. Namely features take values from the interval [0, 1]. In each database, we randomly select half of images in each subject as the training samples, and the rest as the test samples. This procedure is repeated ten times. The polynomial kernel with $d = 3$ is adopted for NNS.

Since the feature number is very large, it is difficult to directly perform on the original data for some methods, such as SRC. The random projection (RP) method is used for reducing the dimensionality of original data. RP shows its effective in face recognition [12]. Eight dimensionality are adopted here, or 10, 20, 30, 40, 50, 70, 100, and 150. For each dimensionality, random projection is performed 10 runs.

ORL Database: There are 10 different images for each subject in the ORL face database composed of 40 distinct subjects. All the subjects are in up-right, frontal position (with tolerance for some side movement). The size of each face image is 112×92, and the resulting standardized input vectors are of dimensionality 10,304. The number of images for both training and test is 200. The classification errors on test set are reported in Figure 2(a). We can see that NNS is the best algorithm from 10D to 150D on the ORL face data, where D denotes dimensionality. In addition, NNS slightly decreases its error when the dimensionality is larger than about 40. Whilst NN and NS improve their performance as increasing dimensionality.

UMIST Database: The UMIST face database is a multi-view database which consists of 574 cropped gray-scale images of 20 subjects, each covering a wide range of poses from profile to frontal views as well as race, gender and appearance. Each image in the

database is resized into 112×92. The total number of the training samples is 290, and that of the test samples is 284. The final results on the UMIST database are shown in Figure 2(b). We observe that NNS is always better than NS from 10D to 150D. Only on 10D and 20D, NNS is worse than NN and SRC.



(a) ORL (b) UMIST

Fig. 2. Classification errors of test set on two face databases (a) ORL, and (b) UMIST

According to the results on two face databases, we have the following conclusions. NNS is always better than NS when the dimensionality is relatively lower, such as less than 50D. In addition, the performance NNS is improved slightly when increasing dimensionality to some degree. On two face data sets, NNS is better than SRC.

4 Conclusion

This paper proposes a nonlinear nearest subspace classifier. NS loses its classification ability when applying it to a general classification task in which samples belonging to different classes have the same orientation. This problem can be solved by mapping the samples into a (RBF) kernel empirical mapping space and then performing NS in this space, which is supported by the experiment on toy data set. Experiments on public face data sets are performed. The performance comparison of NNS with NN, NS, and SRC are taken into account. In the case of high-dimensional face data, NNS with polynomial kernel is better than NS when the dimensionality is lower than 50D.

Although NNS is a promising nonparametric classifier, NNS also has its drawback when processing high-dimensional data. NNS first greatly decreases its test error as increasing dimensionality, and then slightly decreases as increasing dimensionality. In [5], a method for optimizing sensing matrix and sparsifying dictionary is proposed. The sample matrix in NS and NNS can be regarded as the sparsifying dictionary. In the further, the optimization of sample matrix will be taken into account to expectantly improve performance of NNS. In addition, we will take into account the selection of kernel parameters for NNS, which would further improve the performance of NNS.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 60970067, 61033013, 60872135 and 60803098.

References

1. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces versus Fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Special Issue on Face Recognition 19, 711–720 (1997)
2. Candès, E., Romberg, J.: ℓ_1-magic: Recovery of sparse signals via convex programming (October 2005), http://www.acm.caltech.edu/l1magic/
3. Chien, J.T., Wu, C.C.: Discriminant waveletfaces and nearest feature classifiers for face recognition. IEEE Transaction on Pattern Analysis and Machine Intelligence 24(12), 1644–1649 (2002)
4. Cohen, M.C., Paliwal, K.K.: Classyfying microarray cancer datasets using nearest subspace classification. In: Chetty, M., Ngom, A., Ahmad, S. (eds.) Third IAPR International Conference on Pattern Recognition in Bioinformatics. LNCS, vol. 5265, Springer, Heidelberg (2008)
5. Duarte-Carvajalino, J.M., Sapiro, G.: Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization. IEEE Transactions on Image Processing 18(7), 1395–1408 (2009)
6. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. John Wiley & Sons (2000)
7. Graham, D.B., Allinson, N.M.: Characterizing virtual Eigensignatures for general purpose face recognition. In: Face Recognition: From Theory to Applications. NATO ASI Series F, Computer and Systems Sciences, vol. 163, pp. 446–456 (1998)
8. Lee, K.C., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(5), 684–698 (2005)
9. Li, S.Z.: Face recognition based on nearest linear combinations. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 839–844. IEEE Computer Society, Washington, DC, USA (1998)
10. Li, S.Z., Lu, J.: Face recognition using the nearest feature line method. IEEE Transactions on Neural Networks 10(2), 439–443 (1999)
11. Samaria, F.S., Harter, A.C.: Parameterisation of a stochastic model for human face identification. In: Proceedings of the 2nd IEEE International Workshop on Applications of Computer Vision, Sarasota, Florida, pp. 138–142 (December 1994)
12. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(2), 210–226 (2009)
13. Zhang, L., Zhou, W.D., Chang, P.C.: Generalized nonlinear discriminant analysis and its small sample size problems. Neurocomputing 74, 568–574 (2011)
14. Zhang, L., Zhou, W.D., Jiao, L.C.: Hidden space support vector machines. IEEE Transactions on Neural Networks 16(6), 1424–1434 (2004)
15. Zhang, L., Zhou, W.D., Jiao, L.C.: Wavelet support vector machine. IEEE Transactions on Systems, Man, and Cybernetics - Part B 34(1), 34–39 (2004)
16. Zhang, L., Zhou, W.D., Jiao, L.C.: Support vector machines based on the orthogonal projection kernel of father wavelet. International Journal of Computational Intelligence and Applications 5(3), 283–303 (2005)
17. Zhou, W., Zhang, L., Jiao, L.: Hidden Space Principal Component Analysis. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 801–805. Springer, Heidelberg (2006), http://www.springerlink.com/content/2102835260t2m2x6/fulltext.pdf
18. Zhou, X.F., Jiang, W.H., Shi, Y.: Credit risk evaluation by using nearest subspace method. Procedia Computer Science 1, 2443–2449 (2010)

A Novel Framework Based on Trace Norm Minimization for Audio Event Detection

Ziqiang Shi, Jiqing Han, and Tieran Zheng

School of Computer Science and Technology, Harbin Institute of Technology,
P.O. Box 321, Harbin 150001, China
{zqshi,jqhan,zhengtieran}@hit.edu.cn

Abstract. In this paper, a novel framework based on trace norm minimization for audio event detection is proposed. In the framework, both the feature extraction and pattern classifier are made by solving corresponding convex optimization problem with trace norm regularization or under trace norm constraint. For feature extraction, robust principle component analysis (robust PCA) via minimizing a combination of the nuclear norm and the ℓ_1-norm is used to extract matrix representation features which is robust to outliers and gross corruption for audio segments. These matrix representation features are fed to a linear classifier where the weight matrix and bias are learned by solving similar trace norm regularized problems. Experiments on real data sets indicate that this novel framework is effective and noise robust.

Keywords: Audio event detection, Trace norm minimization, Low-rank matrix, Robust principle component analysis, Matrix classification.

1 Introduction

Audio event Detection (AED), a subtask of audio scene analysis [1,2,3], has wide applications. For example, highlight sound effects, such as laugh and applause are usually semantically related with highlight events in general videos, like sports, entertainments, meeting, and home videos.

Most of the AED algorithms resort to the two steps approach, which involves extracting discriminatory features from audio data and feeding them to pattern classifier. Feature commonly exploited for audio event detection can be roughly classified into time domain features, transformation domain features, time-transformation domain features or their combinations [4]. Many of those features are common to AED and speech recognition. Having extracted descriptive features, various machine learning methods are used to provide a final classification of the audio events such as rule-based approaches, Gaussian mixture models, support vector machines, and Bayesian networks [4]. In most previous works, these two steps for AED are always separate and independent. In this work, our aim is to propose a novel unified inherent robust framework for both feature extraction and classifier learning using trace norm regularization.

The trace norm regularization is a principled approach to learn low-rank matrices through convex optimization problems [5]. These similar problems arise

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 646–654, 2011.
© Springer-Verlag Berlin Heidelberg 2011

in many machine learning tasks such as matrix completion [6], multi-task learning [7], robust principle component analysis (robust PCA) [8,9], and matrix classification [10]. In this paper, robust PCA is used to extract matrix representation features for audio segments. Unlike traditional frame based vector features, these matrix features are extracted based on sequences of audio frames. It is believed that in a short duration the signals are contributed by a few factors. Thus it is natural to approximate the frame sequence by robust PCA which assumes that the observed matrices are combinations of low-rank matrices and corruption matrices. Having extracted the robust low-rank matrix feature, almost similar regularization framework based matrix classification approach proposed by Tomioka and Aihara in [10] is used to predict the label. In order to obtain a fast learning convergence, recently proposed accelerated proximal gradient (APG) method [11,12,13] is used to learn the weight matrix and the bias.

The paper is organized as follows: Section 2 presents the extraction of low-rank matrix representation feature. The proposed audio event detection with trace norm regularized matrix classification are introduced in Section 3. Section 4 is devoted to experiments to demonstrate the characteristics and merits of the proposed algorithm. Finally we give some concluding remarks in Section 5.

2 Low-Rank Matrix Representation Features

Over the past decades, a lot research has been done on audio and speech features for AED [1,2,3,4]. Due to convenience and the short-time stationary assumption, these features are mainly in vector form based on frames, although it is believed that features based on longer duration help a lot in decision making. In this paper, in order to build long term features, the consecutive frame signals are made together as rows, then the audio segments become matrices. Generally, it is assumed and believed that the consecutive frame signals are influenced by a few factors, thus these matrices are combinations of low-rank components and noise. Hence, it is natural to approximate these matrices by low-rank matrices. In this work, these approximated low-rank matrices are used as features.

Given an observed data matrix $D \in \mathbb{R}^{m\times n}$, where m is the number of frames and n represents the number of samples in a frame, it is assumed that the matrix can be decomposed as

$$D = A + E, \tag{1}$$

where A is the low-rank component and E is the error or noise matrix. The purpose here is to recover the low-rank component without knowing its rank. For this issue, PCA is a suitable approach since it can find the low-dimensional approximating subspace by forming a low-rank approximation to the data matrix [14]. However, it breaks down under large corruption, even if that corruption affects only a very few of the observation which is often encountered in practice [9]. To solve this problem, the following convex optimization formulation is proposed

$$\min_{A,E\in\mathbb{R}^{m\times n}} \|A\|_* + \lambda\|E\|_1, \text{ subject to } D = A + E, \tag{2}$$

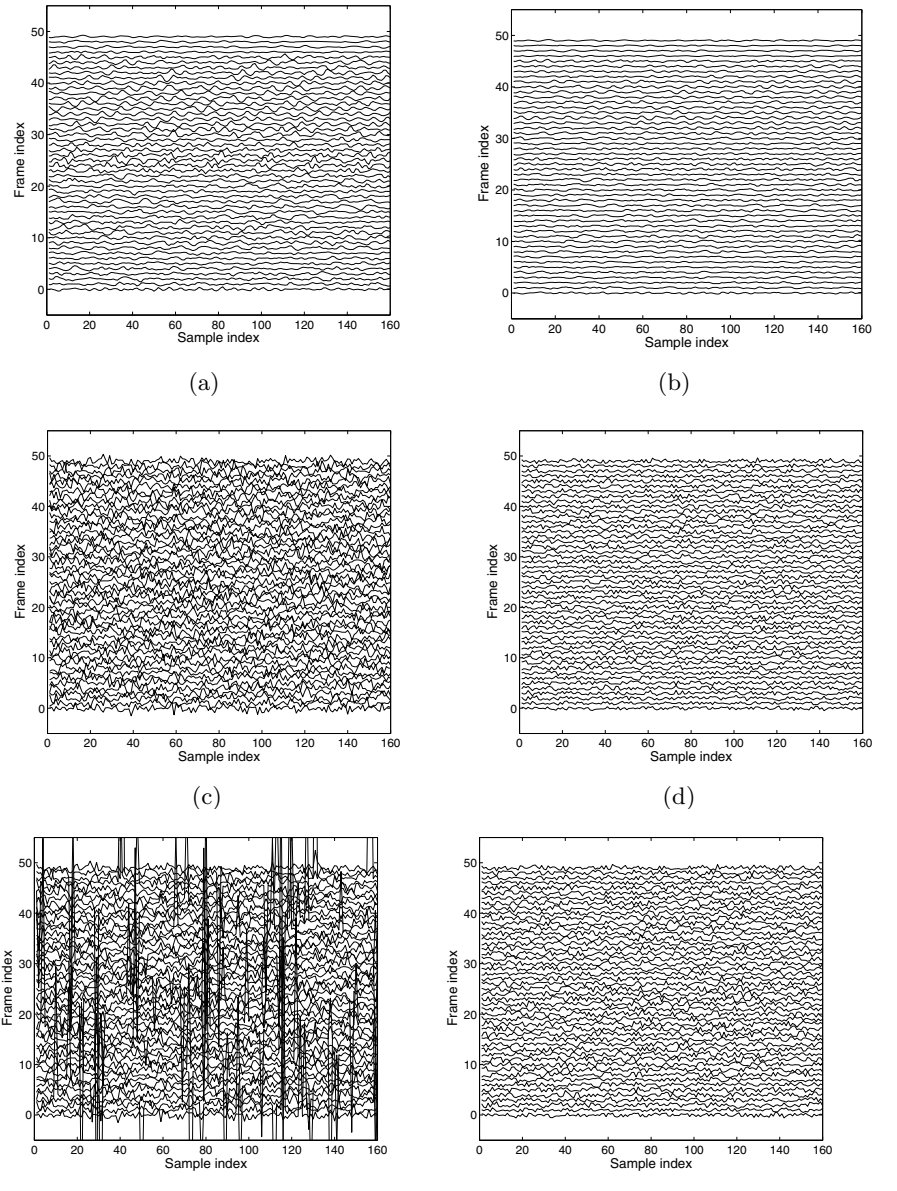


Fig. 1. Matrix form of audio segments with or without noise and extracted matrix features via Robust PCA with $\lambda = 0.1$ throughout. (a) Matrix form of a typical laugh sound effect audio segment; (b) The low-rank component recovered from (a) via robust PCA; (c) Matrix form of the same audio segment corrupted by white Gaussian noise with SNR=20dB; (d) The low-rank component recovered from (c) via robust PCA; (e) Matrix form of the same audio segment corrupted by white Gaussian noise and random large errors; (f) The low-rank component recovered from (e) via robust PCA.

where $\|\cdot\|_*$ denotes the trace norm of a matrix which is defined as the sum of the singular values, $\|\cdot\|_1$ denotes the sum of the absolute values of matrix elements, and λ is a positive regularization parameter. This optimization is refered to as *robust PCA* in [8] for its ability to exactly recover underlying low-rank structure in data even in the presence of large errors or outliers. In order to solve Eq. (2), several algorithms have been proposed, among which the augmented Lagrange multiplier method is the most efficient and accurate so far [9]. In our work, this robust PCA method is employed for the matrix feature extraction.

In order to apply the augmented Lagrange multiplier (ALM) to the robust PCA problem, Lin et. al. [9] identify the problem as

$$X = (A, E), f(X) = \|A\|_* + \lambda\|E\|_1, \text{ and } h(X) = D - A - E, \tag{3}$$

and the Lagrangian function becomes

$$L(A, E, Y, \mu) \doteq \|A\|_* + \lambda\|E\|_1 + < Y, D - A - E > + \frac{\mu}{2}\|D - A - E\|_F^2. \tag{4}$$

Two ALM algorithms to solve the above formulation are proposed in [9]. Considering a balance between processing speed and accuracy, the robust PCA via the inexact ALM method is chosen in our work. Thus the matrix representation feature extraction process based on this approach is summarized in Algorithm 1. In Algorithm 1, $J(D)$ is defined as the larger one of $\|D\|_2$ and $\lambda^{-1}\|D\|_\infty$, where $\|\cdot\|_\infty$ is the maximum absolute value of the matrix elements. The $\mathcal{S}_\varepsilon[\cdot]$ is the soft-thresholding operator introduced in [9].

Figure 1 shows the recovered low-rank matrices via applying robust PCA to the matrix form of a typical laugh sound effect audio segment with or without corruptions. In which, the regularization parameter is fixed as 0.1. It can be seen that robust PCA extracted matrices are robust to large errors and Gaussian noise. Ideally, these above recovered low-rank matrices can be used as features directly. However in practice for such large matrices, the learning process of the next section would take several days, hence in this work the comparative smaller MFCCs (mel-frequency cepstral coefficients) [15] matrices are used instead of the audio wave matrices. MFCCs are extracted for each frame, and several adjacent frames forms a MFCCs matrix. Then robust PCA is used to extract the low-rank components of these MFCCs matrices, and these extracted low-rank components are adopted as features in this work.

3 Matrix Classification and AED

Having extracted robust matrix representation features, the linear matrix classification approach based on trace norm regularization framework [10] is used to classify them. The motivation for trace norm regularization based matrix classification framework is two folds: a) trace norm considers the interactive information among the frames in the matrix while the simple approach that treat the matrix as a long vector would lose the information; b) trace norm is a suitable quantity that measures the complexity of the linear classifier. Generally, the

Algorithm 1. Low-Rank Matrix Representation Feature Extraction for Audio Segments via Robust PCA

Input: $D \in \mathbb{R}^{m\times n}$ (MFCCs matrix of the audio segment).
Initialization: $D \in \mathbb{R}^{m\times n}, Y_0 = D/J(D), E_0 = 0, \mu_0 > 0, k = 0$.
1: **while** not converged **do**
2: // Lines 3-4 solve $A_{k+1} = \arg\min_A L(A, E_k, Y_k, \mu_k)$.
3: $(U, S, V) = \text{svd}(D - E_k + \mu_k^{-1} Y_k)$.
4: $A_k = U\mathcal{S}_{\mu_k^{-1}}[S]V^T$.
5: // Line 6 solves $E_{k+1} = \arg\min_E L(A_{k+1}, E, Y_k, \mu_k)$.
6: $E_{k+1} = \mathcal{S}_{\lambda\mu_k^{-1}}[D - A_{k+1} + \mu_k^{-1} Y_k]$.
7: $Y_{k+1} = Y_k + \mu_k(D - A_{k+1} - E_{k+1})$.
8: Update μ_k to μ_{k+1}.
9: $k \leftarrow k + 1$.
10: **end while**
Output: $W \leftarrow W_k$.

problem for trace norm regularization based matrix classification is formulated as

$$\min_{W,b} F_s(W, b) = \sum_{i=1}^{s} (y_i - \text{Tr}(W^T X_i) - b)^2 + \lambda \|W\|_*, \tag{5}$$

where $W \in \mathbb{R}^{m\times n}$ is the unknown *weight matrix*, $b \in \mathbb{R}$ is the *bias*, and $(X_i, y_i) \in \mathbb{R}^{m\times n} \times \mathbb{R}(i = 1, ..., s)$ are the training samples. Similar with the robust PCA, this framework is also based on trace norm regularization.

In order to obtain the classifier for feature matrices extracted on audio segments, the $O(\frac{1}{k^2})$ converged APG method [11,12,13] is used to solve the above problem, where k is the number of iterations. The learning process based on the APG method of the classifier for the audio segments based matrix features is summarized in Algorithm 2. The general APG algorithms only provide the methods for learning weight matrices, do not give out the bias updating rules. In order to update the bias b, fixes the weight matrix W_k and solve the following problem

$$b_k = \text{argmin}_b \{\sum_{i=1}^{s} (y_i - \text{Tr}(W_k^T X_i) - b)^2 + \lambda \|W_k\|_*\}, \tag{6}$$

which results in the bias updating rule

$$b_k = \frac{1}{s} \sum_{i=1}^{s} (y_i - \text{Tr}(W_k^T X_i)). \tag{7}$$

This results in the line 6 of Algorithm 2. For the stopping criteria of the iterations, we take the following relative error conditions:

$$\|W_{k+1} - W_k\|_F / (\|W_k\|_F + 1) < \varepsilon_1 \text{ and } |b_{k+1} - b_k| / (|b_k| + 1) < \varepsilon_2. \tag{8}$$

Algorithm 2. Learning of Audio Segments Classifier via APG

Input: $(X_i, y_i) \in \mathbb{R}^{m\times n} \times \mathbb{R}(i = 1, ..., s), \lambda$.

Initialization: $W_0 = Z_1 \in \mathbb{R}^{m\times n}, b_0 \in \mathbb{R}, \alpha_1 = 1, L = 2mn \sum_{i=1}^{s} \|X_i\|_F^2$.

1: **while** not converged **do**
2: $(U, S, V) = \text{svd}(Z_k - \frac{1}{L}(-2\sum_{i=1}^{s}(y_i - \text{Tr}(Z_k^T X_i))X_i))$.
3: $W_k = U\mathcal{S}_{\frac{\lambda}{L}}[S]V^T$.
4: $\alpha_{k+1} = \frac{1+\sqrt{1+4\alpha_k^2}}{2}$.
5: $Z_{k+1} = W_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(W_k - W_{k-1})$.
6: $b_k = \frac{1}{s}\sum_{i=1}^{s}(y_i - \text{Tr}(W_k^T X_i))$.
7: $k \leftarrow k + 1$.
8: **end while**
Output: $W \leftarrow W_k, b \leftarrow b_k$.

After the weight matrix W and bias b are found, the observed MFCCs matrix X_i can be classified via

$$\hat{y}_i = \text{Tr}(W^T X_i) + b. \tag{9}$$

Based on these classification results, then the AED can be performed. Let $e_k = (y_i, \cdots, y_{i+\triangle i})$ be an audio event, where $y_i, \cdots, y_{i+\triangle i}$ are the true labels of the short segments in this event. If several adjacent audio segments are classified as event related, then this will be a detected event. In this paper, if

$$|\{j | y_j * \hat{y}_j > 0, i \le j \le i + \triangle i\}| \ge \frac{\triangle i}{2}, \tag{10}$$

that is to say, half of the audio segments in the event are classified right, then the event is detected. Hence the recall and precision of the event detection can be computed.

4 Experimental Validation

Experiments are conducted on a collected database. We downloaded about 20 hours videos from Youku [16], with different programs and different languages. The start and end position of all the applause and laugh of the audio-tracks are manually labeled. The database includes 800 segments of each sound effect. Each segment is about 3-8s long and totally about 1hour data for each sound effect. All the audio recordings were converted to monaural wave format at a sampling frequency of 8kHz and quantized 16bits. Furthermore, the audio signals have been normalized, so that they have zero mean amplitude with unit variance in order to remove any factors related to the recording conditions.

In order to assess the effectiveness of robust PCA extracted low-rank matrix features and the corresponding matrix classification method, detailed experiments are conducted. Original features (MFCCs_Matrix), corrupted with 0dB

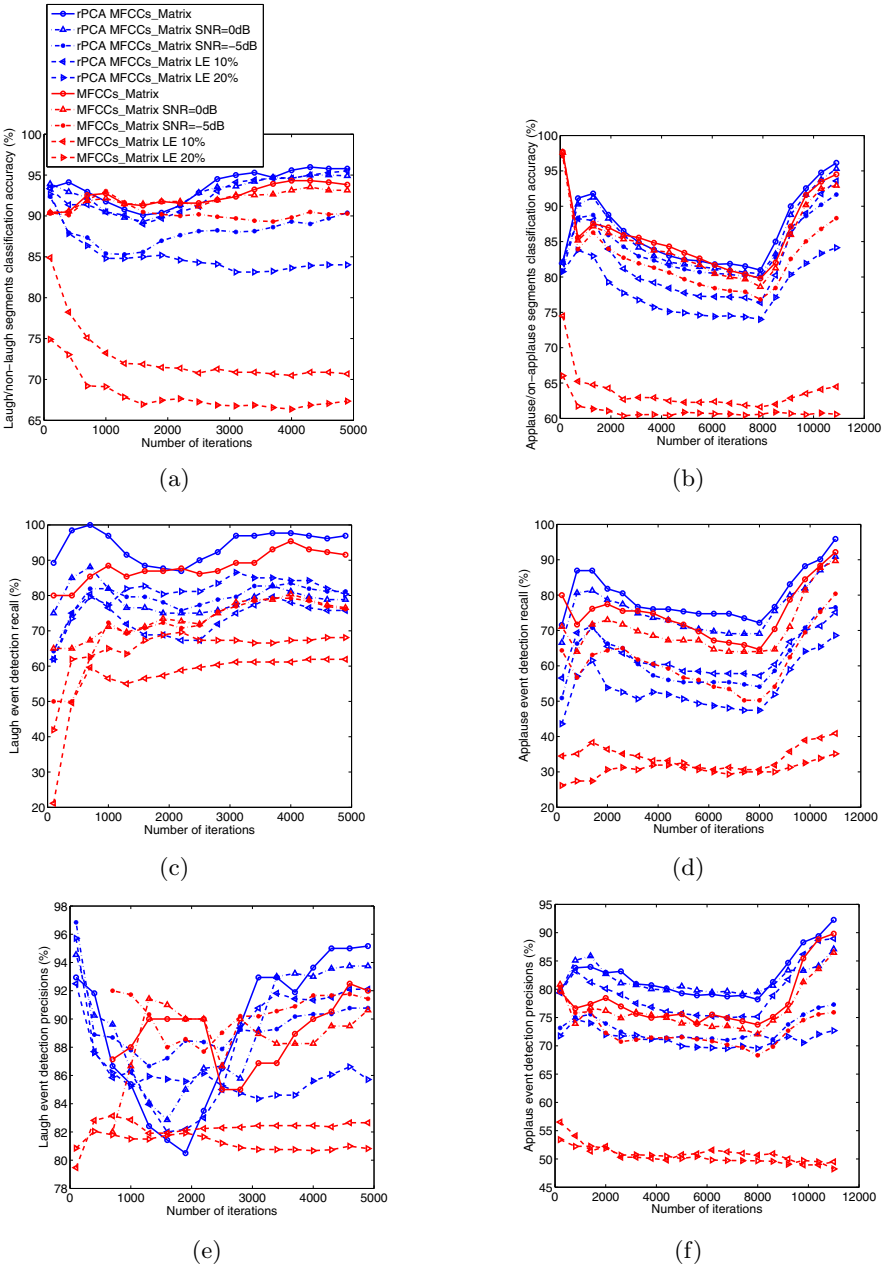


Fig. 2. (a),(c),and (e): Comparisons of robust PCA extracted low-rank features and MFCCs matrices in laugh event detection. (b),(d),and (f): Comparisons of robust PCA extracted low-rank features and MFCCs matrices in applause event detection. Since all the sub figures shares the same legend, and also to save space, so only one legend is showed in this figure (in the sub figure (a)).

and -5dB white Gaussian noise (WGN SNR=0dB, -5dB) and 10%, 20% random large errors (LE 10%, 20%), and parallelism robust PCA extracted features (rPCA) are compared. In the comparisons, the parameters in the stopping criteria Eq. (8) are $\varepsilon_1 = 10^{-6}$ and $\varepsilon_2 = 10^{-6}$. Audio streams were windowed into a sequence of short-term frames (20 ms long) with non overlap. 13 dimensional MFCCs including energy are extracted, and adjacent 50 frames (one second) of MFCCs form the MFCCs matrix feature. The regularization constant λ is set $1/\sqrt{50}$ which is a classical normalization factor according to [17].

Three measurements are used to evaluate the performance of the methods. The first one is the classification accuracy of the one second audio segments obtained in Algorithm 1. The second and third ones are the precision and recall of the events.

Figure 2 shows the performances of the methods with different matrix features under different noise conditions as the functions of the number of iterations used in Algorithm 2. It can be seen that the original MFCCs matrix feature is not robust to noises, especially random large errors. If 10% of the elements of the MFCCs matrix features are corrupted with random large errors, then generally there would be a decrease of 20% in audio segments classification accuracy, 20% in event detection recall, and 10% in event detection precision. While for robust PCA extracted low-rank features, the decreases are 2%, 8%, and 4% respectively. The robust PCA feature is also robust to WGN, since there is almost no decrease in classification accuracy when the original feature is added with 0dB WGN, and only 4% decrease when added with -5dB WGN. The experiments show that the low-rank components are more robust to noises and errors than the original features.

5 Conclusions

In this work, we present a new framework based on trace norm minimization for audio event detection. The novel method unified feature extraction and classifier learning into the same framework. In this framework, robust PCA extracted low-rank component of original signal or feature is more robust to corrupted noise and errors, especially to random large errors. Experiments show that even the percent of the original feature elements corrupted with random large errors is up to 10%, the performance of the robust PCA extracted features almost have no decrease. In future work, we plan to test this robust feature in other audio or speech processing related applications and extend robust PCA, even trace norm minimization related methods from matrices to the more general multi-way arrays (tensors).

Acknowledgments. This work was supported by the grant from the National Basic Research Program of China (973 Program) No. 2007CB311100 and National Natural Science Foundation of China (No. 61071181).

References

1. Lu, L.: Content analysis for audio classification and segmentation. IEEE Trans. Speech and Audio Processing 10, 504–516 (2002)
2. Cui, R., Lu, L., Zhung, H.J., Cai, L.H.: Highlight sound effects detection in audio stream. In: Proceedings of IEEE International Conference on Multimedia and Expo, pp. 37–40 (2003)
3. Pradeep, K.A., Namunu, C.M., Mohan, S.K.: Audio based event detection for multimedia surveillance. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (2006)
4. Zhuang, X., Zhou, X., Huang, T.S., Hasegawa-Johnson, M.: Feature analysis and selection for acoustic event detection. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 17–20 (2008)
5. Fazel, M., Hindi, H., Boyd, S.P.: A rank minimization heuristic with application to minimum order system approximation. In: Proceedings of the American Control Conference, pp. 4734–4739 (2001)
6. Srebro, N., Rennie, J.D.M., Jaakkola, T.S.: Maximum-margin matrix factorization. In: Proceedings of Advances in Neural Information Processing Systems, pp. 1329–1336 (2005)
7. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Machine Learning 73(3), 243–272 (2008)
8. Wright, J., Ganesh, A., Rao, S., Peng, Y., Ma, Y.: Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In: Proceedings of Advances in Neural Information Processing Systems (2009)
9. Lin, Z., Chen, M., Wu, L., Ma, Y.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. UIUC Technical Report (2009)
10. Tomioka, R., Aihara, K.: Classifying matrices with a spectral regularization. In: 24th International Conference on Machine Learning, pp. 895–902 (2007)
11. Toh, K., Yun, S.: An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. Pacific J. Optim. 6, 615–640 (2010)
12. Ji, S., Ye, J.: An accelerated gradient method for trace norm minimization. In: 26th International Conference on Machine Learning, pp. 457–464 (2009)
13. Liu, Y.J., Sun, D., Toh, K.C.: An implementable proximal point algorithmic framework for nuclear norm minimization. Mathematical Programming, 1–38 (2009)
14. Jolliffe, I.T.: Principal Component Analysis. Springer Series in Statistics. Springer, Berlin (1986)
15. Davis, S., Mermelstein, P.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Transactions on Acoustics, Speech and Signal Processing 28(4), 357–366 (1980)
16. Youku, http://www.youku.com
17. Bickel, P., Ritov, Y., Tsybakov, A.: Simultaneous analysis of Lasso and Dantzig selector. Annals of Statistics 37(4), 1705–1732 (2009)

A Modified Multiplicative Update Algorithm for Euclidean Distance-Based Nonnegative Matrix Factorization and Its Global Convergence

Ryota Hibi[1] and Norikazu Takahashi[1,2]

[1] Department of Informatics, Kyushu University,
744 Motooka, Nishi-ku, Fukuoka 819-0395 Japan
[2] Institute of Systems, Information Technologies and Nanotechnologies,
2-1-22 Momochihama, Sawara-ku, Fukuoka 814-0001, Japan
hibi@kairo.csce.kyushu-u.ac.jp, norikazu@inf.kyushu-u.ac.jp

Abstract. Nonnegative matrix factorization (NMF) is to approximate a given large nonnegative matrix by the product of two small nonnegative matrices. Although the multiplicative update algorithm is widely used as an efficient computation method for NMF, it has a serious drawback that the update formulas are not well-defined because they are expressed in the form of a fraction. Furthermore, due to this drawback, the global convergence of the algorithm has not been guaranteed. In this paper, we consider NMF in which the approximation error is measured by the Euclidean distance between two matrices. We propose a modified multiplicative update algorithm in order to overcome the drawback of the original version and prove its global convergence.

Keywords: nonnegative matrix factorization, multiplicative update, Euclidean distance, global convergence.

1 Introduction

Nonnegative matrix factorization (NMF) [1,2] is to approximate a given large nonnegative matrix by the product of two small nonnegative matrices. Since it can not only reduce the amount of data but also find nonnegative basis for the given nonnegative data, NMF has been applied to various problems in machine learning, signal processing, and so on [1,3,4,5,6].

In general, NMF is formulated as a constrained optimization problem in which the approximation error has to be minimized with respect to two factor matrices subject to the nonnegativity of those matrices. Lee and Seung [2] considered NMF problems in which the approximation error is measured by the Euclidean distance or the divergence, and proposed an iterative method called the multiplicative update algorithm.

Although the multiplicative update algorithm is widely used as an efficient computation method for NMF, it has a serious theoretical drawback that the update formulas are not well-defined because they are expressed in the form of a fraction. Furthermore, due to this drawback, the global convergence has not been

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 655–662, 2011.
© Springer-Verlag Berlin Heidelberg 2011

guaranteed. By global convergence, we mean that the algorithm converges to a stationary point of the optimization problem for any initial condition. Therefore, the global convergence analysis of the multiplicative update algorithm is an important research topic in NMF, and many papers have been published so far [7,8,9]. Lin [8] considered the case of the Euclidean distance minimization and showed that some modifications to the original algorithm by Lee and Seung can make it well-defined and globally convergent. However, since Lin's modified algorithm is not multiplicative but additive, the result cannot be directly applied to the original algorithm. Finesso and Spreij [7] considered the case of the divergence minimization and derived some theoretical results about its stability properties. However, ill-definedness of the update rule still remains in their analysis. Recently, Badeau *et al.* [9] studied local stability of the multiplicative update algorithm by using Lyapunov methods and showed that the local optimal solution is asymptotically stable if one of two factor matrices is fixed.

In this paper, we study the multiplicative update algorithm in which the Euclidean distance is minimized, and show that a minimal modification to the original algorithm [2] makes it well-defined and globally convergent. The only difference between the original and modified algorithms is that the latter does not allow variables to take values less than the user-specified positive constant. Unlike the algorithms of Lin [8] and Finesso and Spreij [7], our algorithm does not require a normalization procedure. We will prove the global convergence of the modified algorithm by using Zangwill's global convergence theorem [10], which is well known in optimization theory and has played important roles in the field of machine learning [11,12].

2 Multiplicative Update by Lee and Seung

Given a nonnegative matrix $V \in \mathbb{R}^{n\times m}$ and a positive integer r, nonnegative matrix factorization (NMF) is to find two nonnegative matrices $W \in \mathbb{R}^{n\times r}$ and $H \in \mathbb{R}^{r\times m}$ such that

$$V \approx WH\,.$$

Let us consider each column of V as a data vector. If the value of r is sufficiently small, NMF gives us a compact expression for the original data because the total number of elements of two matrices W and H is less than that of the data matrix V. Moreover, the columns of W are regarded as a kind of basis for the space spanned by the columns of V because each data vector can be approximated by a linear combination of the columns of W.

Lee and Seung [2] employed the Euclidean distance and the divergence for the approximation error between V and WH, and formulated NMF as two types of optimization problems in which the approximation error should be minimized under the constraint that W and H are nonnegative. In the case of the Euclidean distance, the optimization problem is expressed as follows:

$$\begin{array}{l}\text{minimize} \quad f(W,H) = \|V - WH\|^2 \\ \text{subject to } H_{aj} \geq 0,\ W_{ia} \geq 0,\ \forall a,i,j\end{array} \tag{1}$$

where $\|\cdot\|$ represents the Frobenius norm, that is,

$$\|V - WH\|^2 = \sum_{ij} (V_{ij} - (WH)_{ij})^2 ,$$

H_{aj} denotes the (a, j) element of H, and W_{ia} denotes the (i, a) element of W.

In general, it is difficult to find an optimal solution of the problem (1) because the objective function $f(W, H)$ is not convex. Therefore, we have to take the second best way, that is, we try to find a local optimal solution. For this purpose, Lee and Seung [2] proposed the update rules [1]:

$$H_{aj}^{k+1} = H_{aj}^{k} \frac{((W^k)^T V)_{aj}}{((W^k)^T W^k H^k)_{aj}} \tag{2}$$

$$W_{ia}^{k+1} = W_{ia}^{k} \frac{(V(H^{k+1})^T)_{ia}}{(W^k H^{k+1} (H^{k+1})^T)_{ia}} \tag{3}$$

as an efficient method for finding a local optimal solution of (1). Eqs.(2) and (3) are called the multiplicative updates because a new estimate is expressed as the product of a current estimate and some factor.

As a preparation for the following sections, we now review how the multiplicative updates are derived, but notations used here are different from the original paper [2]. First, we consider the problem of minimizing $f(W^*, H)$ where W^* is any positive constant matrix. A function $g(W^*, H, H')$ satisfying

$$g^{W^*}(H, H') \geq f(W^*, H), \quad \forall H, H' > 0 \tag{4}$$

$$g^{W^*}(H, H) = f(W^*, H), \quad \forall H > 0 \tag{5}$$

is called an auxiliary function of $f(W^*, H)$. If we update the matrix H by

$$H^{k+1} = \arg\min_{H} g^{W^k}(H, H^k)$$

then the value of the objective function $f(W, H)$ decreases or remains the same because the inequalities

$$f(W^k, H^{k+1}) \leq g^{W^k}(H^{k+1}, H^k) \leq g^{W^k}(H^k, H^k) = f(W^k, H^k) \tag{6}$$

hold due to (4) and (5). As a candidate for an auxiliary function of $f(W^*, H)$, let us consider

$$g^{W^*}(H, H') = f(W^*, H') + \sum_{a=1}^{r} \sum_{j=1}^{m} g_{aj}^{W^*}(H_{aj}, H')$$

where $g_{aj}^{W^*}(H_{aj}, H')$ is defined by

$g_{aj}^{W^*}(H_{aj}, H')$

[1] Although it is not clearly written in [2] which of H^k and H^{k+1} is used for the computation of W^{k+1}, we use the latter for our later discussion.

$$= (\nabla_H f(W^*, H'))_{aj}(H_{aj} - H'_{aj}) + \frac{((W^*)^T W^* H')_{aj}}{H'_{aj}}(H_{aj} - H'_{aj})^2$$

$$= \left((W^*)^T(W^*H' - V)\right)_{aj}(H_{aj} - H'_{aj}) + \frac{((W^*)^T W^* H')_{aj}}{H'_{aj}}(H_{aj} - H'_{aj})^2 . \quad (7)$$

It is apparent that the function $g^{W^*}(H, H')$ satisfies the condition (5). Also, it is not so difficult to show that $g^{W^*}(H, H')$ satisfies the condition (4).

Let W^k and H^k be given positive matrices. In order to minimize $g^{W^k}(H, H^k)$ with respect to H, it suffices for us to minimize $g_{aj}^{W^k}(H_{aj}, H^k)$ $(a = 1, 2, \ldots, r;$ $j = 1, 2, \ldots, m)$ with respect to H_{aj} independently. Furthermore, since the function $g_{aj}^{W^k}(H_{aj}, H^k)$ is strictly convex with respect to H_{aj}, the equation $\partial g_{aj}^{W^k}(H_{aj}, H^k)/\partial H_{aj} = 0$ has a unique solution and it is the global minimum point of $g_{aj}^{W^k}(H_{aj}, H^k)$. If fact, by putting

$$\frac{\partial g_{aj}^{W^k}(H_{aj}, H^k)}{\partial H_{aj}}$$

$$= \left((W^k)^T(W^kH^k - V)\right)_{aj} + \frac{((W^k)^T W^k H^k)_{aj}}{H'_{aj}}(H_{aj} - H^k_{aj})$$

$$= -\left((W^k)^T V\right)_{aj} + \frac{((W^k)^T W^k H^k)_{aj}}{H^k_{aj}} H_{aj}$$

equal to zero and solving it for H_{aj}, we can derive the update rule (2).

Next, we consider the problem of minimizing $f(W, H^*)$ where H^* is any positive constant matrix. A function $h^{H^*}(W, W')$ satisfying

$$h^{H^*}(W, W') \geq f(W, H^*), \quad \forall W, W' > 0 \quad (8)$$

$$h^{H^*}(W, W) = f(W, H^*), \quad \forall W > 0 \quad (9)$$

is called an auxiliary function of $f(W, H^*)$. If we update the matrix W by

$$W^{k+1} = \arg\min_W h^{H^{k+1}}(W, W^k)$$

then the value of the objective function $f(W, H)$ decreases or remains the same because the inequalities

$$f(W^{k+1}, H^{k+1}) \leq h^{H^{k+1}}(W^{k+1}, W^k) \leq h^{H^{k+1}}(W^k, W^k) = f(W^k, H^{k+1}) \quad (10)$$

hold due to (8) and (9). As a candidate for an auxiliary function of $f(W, H^*)$, let us now consider

$$h(W, W', H^*) = f(W, H^*) + \sum_{i=1}^{n}\sum_{a=1}^{r} h_{ia}^{H^*}(W_{ia}, W')$$

where $h_{ia}^{H^*}(W_{ia}, W')$ is defined by

$$h_{ia}^{H^*}(W_{ia}, W')$$

$$= (\nabla_W f(W', H^*))_{ia}(W_{ia} - W'_{ia}) + \frac{(W'H^*(H^*)^T)_{ia}}{W'_{ia}}(W_{ia} - W'_{ia})^2$$

$$= \left((W'H^* - V)(H^*)^T\right)_{ia}(W_{ia} - W'_{ia}) + \frac{(W'H^*(H^*)^T)_{ia}}{W'_{ia}}(W_{ia} - W'_{ia})^2 \quad (11)$$

It is apparent that $h^{H^*}(W, W')$ satisfies (9). Moreover, we can show that $h^{H^*}(W, W')$ also satisfies (8) in the same way as $g^{W^*}(H, H')$.

Let W^k and H^{k+1} be given positive matrices. In order to minimize the function $h^{H^{k+1}}(W, W^k)$ with respect to W, it suffices for us to minimize $h_{ia}^{H^{k+1}}(W_{ia}, W^k)$ $(i = 1, 2, \ldots, r; a = 1, 2, \ldots, n)$ with respect to W_{ia} independently. Furthermore, since $h_{ia}^{H^{k+1}}(W_{ia}, W^k)$ is strictly convex with respect to W_{ia}, the equation $\partial h_{ia}^{H^{k+1}}(W_{ia}, W^k)/\partial W_{ia} = 0$ has a unique solution and it is the global minimum point of $h_{ia}^{H^{k+1}}(W_{ia}, W^k)$. In fact, by putting

$$\frac{\partial h_{ia}^{H^{k+1}}(W_{ia}, W^k)}{\partial W_{ia}}$$

$$= \left((W^k H^{k+1} - V)(H^{k+1})^T\right)_{ia} + \frac{(W^k H^{k+1}(H^{k+1})^T)_{ia}}{W'_{ia}}(W_{ia} - W^k_{ia})$$

$$= -\left(V(H^{k+1})^T\right)_{ia} + \frac{(W^k H^{k+1}(H^{k+1})^T)_{ia}}{W^k_{ia}} W_{ia}$$

equal to zero and solving it for W_{ia}, we can derive the update rule (3).

3 Modifications to Multiplicative Update Rule and Optimization Problem

The most serious problem in the multiplicative update rule described by (2) and (3) is that the right-hand sides are not defined for all nonnegative matrices W^k and H^k. For example, in the case where either $W^k = 0$ or $H^k = 0$, we cannot obtain the value of H^{k+1} as the denominator in (2) vanishes. In order to avoid this problem, we employ the update rule

$$H_{aj}^{k+1} = \max\left(H_{aj}^k \frac{((W^k)^T V)_{aj}}{((W^k)^T W^k H^k)_{aj}}, \epsilon\right) \quad (12)$$

$$W_{ia}^{k+1} = \max\left(W_{ia}^k \frac{(V(H^{k+1})^T)_{ia}}{(W^k H^{k+1}(H^{k+1})^T)_{ia}}, \epsilon\right) \quad (13)$$

instead of (2) and (3), where ϵ is any positive constant. Equation (12) means that we set H_{aj}^{k+1} to the global minimum point of $g_{aj}^{W^k}(H_{aj}, H^k)$ if it is greater than ϵ and to ϵ otherwise. Similarly, (13) means that we set W_{ia}^{k+1} to the global minimum point of $h_{ia}^{H^{k+1}}(W_{ia}, W^k)$ if it is greater than ϵ and to ϵ otherwise. Therefore, if we define the set X as

$$X = \{(W, H) \,|\, W_{ia} \geq \epsilon,\ H_{aj} \geq \epsilon,\ \forall i, a, j\}$$

then the following lemma holds obviously.

Lemma 1. *Let $\{(W^k, H^k)\}_{k=0}^{\infty}$ be any sequence generated by the modified updates (12) and (13) with the initial condition $(W^0, H^0) \in X$. Then $(W^k, H^k) \in X$ for any positive integer k.*

By making use of the strict convexity of the functions $g_{aj}^{W^k}(H_{aj}, H^k)$ and $h_{ia}^{H^{k+1}}(W_{ia}, W^k)$, we easily derive the following lemma.

Lemma 2. *The right-hand side of (12) is the unique optimal solution of the following optimization problem.*

$$\begin{aligned} &\textit{minimize } g^{W^k}(H, H^k) \\ &\textit{subject to } H_{aj} \geq \epsilon, \quad \forall a, j \end{aligned}$$

Also, the right-hand side of (13) is the unique optimal solution of the following optimization problem.

$$\begin{aligned} &\textit{minimize } h^{H^{k+1}}(W, W^k) \\ &\textit{subject to } W_{ia} \geq \epsilon, \quad \forall i, a \end{aligned}$$

It follows from Lemmas 1 and 2 that the modified update rule also satisfies both (6) and (10). Therefore, we have the following lemma.

Lemma 3. *Let $\{(W^k, H^k)\}_{k=0}^{\infty}$ be any sequence generated by the modified updates (12) and (13) with the initial condition $(W^0, H^0) \in X$. Then we have $f(W^{k+1}, H^{k+1}) \leq f(W^k, H^k)$ for any nonnegative integer k.*

The modified update rule described by (12) and (13) corresponds to modifying the optimization problem (1) as

$$\begin{aligned} &\text{minimize} \quad f(W, H) = \|V - WH\|^2 \\ &\text{subject to } W_{ia} \geq \epsilon, \ H_{aj} \geq \epsilon, \quad \forall i, a, j \end{aligned} \tag{14}$$

Karush-Kuhn-Tucker condition for this optimization problem is expressed as follows[2]

$$\nabla_W f(W, H) \geq 0 \tag{15}$$

$$\nabla_H f(W, H) \geq 0 \tag{16}$$

$$(\nabla_W f(W, H))_{ia}(\epsilon - W_{ia}) = 0, \quad \forall i, a \tag{17}$$

$$(\nabla_H f(W, H))_{aj}(\epsilon - H_{aj}) = 0, \quad \forall a, j \tag{18}$$

where

$$\begin{aligned} \nabla_W f(W, H) &= 2(WH - V)H^T \\ \nabla_H f(W, H) &= 2W^T(WH - V) \end{aligned}$$

Therefore, a necessary condition for a point (W, H) belonging to the feasible region X of the optimization problem (14) to be a local optimal solution is that the conditions (15)–(18) are satisfied. Hereafter, we call a point $(W, H) \in X$ a stationary point of (14) if it satisfies (15)–(18), and denote the set of all stationary points by S.

[2] The conditions (15)–(18) are derived by eliminating Lagrange multipliers in the original Karush-Kuhn-Tucker condition.

4 Global Convergence of Modified Algorithm

In this section, we will prove the global convergence of the algorithm described by (12) and (13) by using Zangwill's global convergence theorem [10]. For the notational simplicity, we hereafter express (12) and (13) as $H^{k+1} = A_1(W^k, H^k)$ and $W^{k+1} = A_2(W^k, H^{k+1})$, respectively. Furthermore, we express these two updates by a single formula as

$$(W^{k+1}, H^{k+1}) = A(W^k, H^k) = (A_2(W^k, A_1(W^k, H^k)), A_1(W^k, H^k)) .$$

Zangwill's global convergence theorem says that if a mapping $A : X \to X$ satisfies the following three conditions then the limit of any convergent subsequence of any sequence $\{(W^k, H^k)\}_{k=0}^{\infty}$ generated by A is a stationary point of the optimization problem (14).

1. Any sequence $\{(W^k, H^k)\}_{k=0}^{\infty}$ generated by the mapping A belongs to a compact set in X.
2. There is a function $z : X \to \mathbb{R}$ satisfying the following two conditions.
   (a) If $(W^*, H^*) \notin S$ then $z(A(W^*, H^*))) < z(W^*, H^*)$.
   (b) If $(W^*, H^*) \in S$ then $z(A(W^*, H^*)) \leq z(W^*, H^*)$.
3. The mapping A is continuous outside S.

It is obvious from the definition of the updates (12) and (13) that the mapping A is continuous. Furthermore, the mapping A also satisfies the remaining two conditions, as shown in the following lemmas.

Lemma 4. *If $(W^*, H^*) \notin S$ then $f(A(W^*, H^*)) < f(W^*, H^*)$ holds. Also, if $(W^*, H^*) \in S$ then $A(W^*, H^*) = (W^*, H^*)$ holds, that is, S is identical with the set of fixed points of the mapping A.*

Lemma 5. *Any sequence $\{(W^k, H^k)\}_{k=0}^{\infty}$ generated by the mapping A with the initial condition $(W^0, H^0) \in X$ belongs to a compact set in X.*

We have shown that all of the three conditions in Zangwill's global convergence theorem are satisfied. Therefore, as a main result of this paper, we obtain the following theorem.

Theorem 1. *Let $\{(W^k, H^k)\}_{k=0}^{\infty}$ be any sequence generated by the modified updates (12) and (13) with the initial condition $(W^0, H^0) \in X$. Then the sequence has at least one convergent subsequence and the limit of any convergent subsequence is a stationary point of the optimization problem (14).*

5 Conclusion

We have proposed a modified multiplicative update algorithm for the Euclidean distance-based NMF and proved that it is globally convergent. Although it was not shown in this paper due to limitations of space, we can easily construct an algorithm that terminates within a finite number of iterations, by further modifying the proposed algorithm. One of the future problems is to consider the case of the divergence minimization.

Acknowledgments. This work was partially supported by Grant-in-Aid for Scientific Research (C) 21560068 from the Japan Society for the Promotion of Science (JSPS), and by the project "R&D for cyber-attack predictions and rapid response technology by means of international cooperation" of the Ministry of Internal Affairs and Communications, Japan.

References

1. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–792 (1999)
2. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Leen, T.K., Dietterich, T.G., Tresp, V. (eds.) Advances in Neural Information Processing Systems, vol. 13, pp. 556–562 (2001)
3. Brunet, J.P., Tamayo, P., Golub, T.R., Mesirov, J.P.: Metagenes and molecular pattern discovery using matrix factorization. Proceedings of National Academy of Science 101(12), 4164–4169 (2004)
4. Berry, M.W., Browne, M.: Email surveillance using non-negative matrix factorization. Computational and Mathematical Organization Theory 11, 249–264 (2005)
5. Holzapfel, A., Stylianou, Y.: Musical genre classification using nonnegative matrix factorization-based features. IEEE Transactions on Audio, Speech, and Language Processing 16(2), 424–434 (2008)
6. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.I.: Nonnegative Matrix and Tensor Factorizations. John Wiley & Sons, West Sussex (2009)
7. Finesso, L., Spreij, P.: Nonnegative matrix factorization and I-divergence alternating minimization. Linear Algebra and its Applications 416, 270–287 (2006)
8. Lin, C.J.: On the convergence of multiplicative update algorithms for nonnegative matrix factorization. IEEE Transactions on Neural Networks 18(6), 1589–1596 (2007)
9. Badeau, R., Bertin, N., Vincent, E.: Stability analysis of multiplicative update algorithms and application to nonnegative matrix factorization. IEEE Transactions on Neural Networks 21(12), 1869–1881 (2010)
10. Zangwill, W.I.: Nonlinear programming: A unified approach. Prentice-Hall (1969)
11. Wu, C.F.J.: On the convergence properties of the EM algorithm. The Ananls of Statistics 11(1), 95–103 (1983)
12. Takahashi, N.: Global convergence of decomposition learning methods for support vector machines. IEEE Transactions on Neural Networks 17(6), 1362–1369 (2006)

A Two Stage Algorithm for K-Mode Convolutive Nonnegative Tucker Decomposition

Qiang Wu[1], Liqing Zhang[2], and Andrzej Cichocki[3]

[1] School of Information Science and Engineering, Shandong University, Jinan, Shandong, China
[2] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[3] Laboratory for Advanced Brain Signal Processing, BSI RIKEN, Wakoshi, Saitama, Japan
johnstonwu@gmail.com, lqzhang@sjtu.edu.cn, cia@brain.riken.jp

Abstract. Higher order tensor model has been seen as a potential mathematical framework to manipulate the multiple factors underlying the observations. In this paper, we propose a flexible two stage algorithm for K-mode Convolutive Nonnegative Tucker Decomposition (K-CNTD) model by an alternating least square procedure. This model can be seen as a convolutive extension of Nonnegative Tucker Decomposition (NTD). Shift-invariant features in different subspaces can be extracted by the K-CNTD algorithm. We impose additional sparseness constraint on the algorithm to find the part-based representations. Extensive simulation results indicate that the K-CNTD algorithm is efficient and provides good performance for a feature extraction task.

1 Introduction

Tensor factorization methods are frequently used in many fields including signal processing, machine learning, computer vision and neuroscience[1]. Various algorithms have been proposed for the factorization models such as the PARAFAC and Tucker models. Compared to traditional subspace methods, tensor factorization models can preserve the natural structure of higher order data without matricizing or vectorizing and provide the unique optimal solutions without imposing orthogonal or independent constraint.

Common tensor factorization methods include PARAFAC model[2], Tucker model[3], Nonnegative Tensor Factorization[1] which imposes the nonnegative constraint on the PARAFAC or Tucker model. Various tensor factorization models beyond PARAFAC and Tucker models have been proposed over years such as INDSCAL, DEDICOM [1] exploring symmetry in tensors and Block Term Decomposition and CANDELING considering models interpolating between PARAFAC and Tucker models. De Lathauwer [4] proposed the higher-order singular value decomposition for tensor decomposition, which is a multilinear generalization of the matrix SVD.

Recently, several papers [5,6] investigate the degeneracy problem of tensor factorization caused by component delays. The shifted or convolutive tensor

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 663–670, 2011.
© Springer-Verlag Berlin Heidelberg 2011

factorization model can been seen as an extension of original tensor factorization to suit with the practical data better. As stated in [5,6,7], there are potential applications to fMRI data, EEG data, astronomical spectrometers for shifted or convolutive tensor factorization methods.

In this paper we propose a two stage algorithm for K-mode Convolutive Nonnegative Tucker Decomposition (K-CNTD) model which can be seen as a generalization of NTD. We first employ ALS NTD to factorize the convolutive mixture in tensor structure into factor matrices and core tensor. Then the original components in K modes are recovered by the ALS convolutive NMF algorithms with sparse constraint. We test the performance of K-CNTD algorithm by the synthetic data and real data. Experimental results show that our proposed algorithm provide good performance for shift-invariant feature extraction with application on speaker recognition in noisy conditions.

2 Two Stage Algorithm for K-Mode Convolutive Nonnegative Tucker Decomposition

2.1 ALS Convolutive NMF with Sparse Constraint

For certain kinds of data, the relative position of basis functions or coefficients in feature space give important meaning. We define the upward, downward, left and right shifted operator on the matrix A by shifting and zero padding the rows or columns of A as following:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \overset{1\rightarrow}{A} = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 4 & 5 \\ 0 & 7 & 8 \end{pmatrix} \overset{1\leftarrow}{A} = \begin{pmatrix} 2 & 3 & 0 \\ 5 & 6 & 0 \\ 8 & 9 & 0 \end{pmatrix} \overset{1\uparrow}{A} = \begin{pmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \\ 0 & 0 & 0 \end{pmatrix} \overset{1\downarrow}{A} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

Paris Smaragdis proposed a convolutive extension to NMF [8] that aims to extract cross-column patterns as single basis function. The convolutive model is defined as

$$V \approx \sum_{l=0}^{L-1} W_l \overset{l\rightarrow}{H} \tag{1}$$

where $V \in \mathbb{R}^{M \times N} \geq 0$ is the input matrix, $W_l|_{l=0}^{L-1} \in \mathbb{R}^{M \times R} \geq 0$ is a set of basis functions and $H \in \mathbb{R}^{R \times N}$ is the coefficients.

Above model (1) can be decomposed into a set of NMF approximations because this is a linear model [8]. Using the alternating least square method described in [9], the ALS update rules of each NMF approximation for W_l and $\overset{l\rightarrow}{H}$ can be derived as

$$H_l \leftarrow \left[\left(W_l^T W_l \right)^{-1} \left(W_l^T \overset{l\leftarrow}{V} - \alpha \mathbf{1} \right) \right]_+, \quad W_l \leftarrow \left[V \overset{l\rightarrow}{H}^T \left(\overset{l\rightarrow}{H} \overset{l\rightarrow}{H}^T \right)^{-1} \right]_+ \tag{2}$$

where $(\cdot)^T$ is the transpose operator, $[a]_+ = max(\epsilon, a)$ is a half-wave rectifying nonlinear projection to enforce nonnegativity[9], $\alpha \geq 0$ is the regularization parameter controlling level of sparsity, for each l, H_l corresponds to $\overset{l\rightarrow}{H}$. In every iteration, the basis function W_l and coefficient matrix H_l are updated for each l. As stated in [8], the algorithm first update all W_l and then H is assigned to the average of $H_l|_{l=0}^{L-1}$, i.e. $H \leftarrow \frac{1}{L}\sum_{l=0}^{L-1} H_l$. The algorithm for ALS Convolutive NMF (CNMF) is described in Algorithm.1.

Algorithm 1. ALS Convolutive NMF with Sparse Constraint

Data: Given data matrix $V \in \mathbb{R}^{M\times N}$, the component number R, the convolutive length L, regularization parameter α.
Result: The estimated matrices $W_l|_{l=0}^{L-1} \in \mathbb{R}^{M\times R}$ and $H \in \mathbb{R}^{R\times N}$.
Initialization: Set $W^{(l)}|_{l=0}^{L-1}$ and H randomly;
repeat
  for $l = 0 : L-1$ **do**
    $H_l \leftarrow \left[\left(W_l^T W_l\right)^{-1}\left(W_l^T \overset{l\leftarrow}{V} - \alpha\mathbf{1}\right)\right]_+$; $W_l \leftarrow \left[V \overset{l\rightarrow^T}{H}\left(\overset{l\rightarrow}{H}\overset{l\rightarrow^T}{H}\right)^{-1}\right]_+$;
  $H \leftarrow \frac{1}{L}\sum_{l=0}^{L-1} H_l$
until *convergence criterion is reached* ;

2.2 Multilinear Algebra and ALS Nonnegative Tucker Decomposition

Multilinear algebra is the algebra of higher order tensors. A tensor is a higher order generalization of matrix. Let $\underline{X} \in \mathbb{R}^{I_1\times I_2\cdots\times I_N}$ denote a tensor. The order of $\underline{X}$ is N. The mode-n matricization of an N order tensor $\underline{X}$ rearranges the elements of $\underline{X}$ to form the matrix $X_{(n)} \in \mathbb{R}^{I_n\times I_{n+1}I_{n+2}\cdots I_N I_1\cdots I_{n-1}}$. The Kronecker, Hardamard products and element-wise division are denoted respectively by $\otimes, \circledast, \oslash$. The mode-$n$ matrix product defines multiplication of a tensor with a matrix in mode n as $\underline{X} = \underline{G}\times_n U$, where $U \in \mathbb{R}^{I_n\times J}$. In this paper we simplify the notation as

$$\underline{G}\times_1 U^{(1)}\cdots\times_N U^{(n)} = \underline{G}\prod_{n=1}^{N}\times_n U^{(n)}, \tag{3}$$

The details about tensor factorization can be found in [1,4,9].

Nonnegative Tucker Decomposition (NTD) [10] is a special kind of Tucker model with nonnegative constraints. The decomposition model is defined as:

$$\underline{X} = \underline{G}\times_1 U^{(1)}\times_2 U^{(2)}\cdots\times_N U^{(N)} + \underline{E} \tag{4}$$

where $\underline{X} \in \mathbb{R}_+^{I_1\times I_2\cdots\times I_N} \geq 0$ is the data tensor, $\underline{G} \in \mathbb{R}_+^{J_1\times\cdots\times J_N} \geq 0$ is the core tensor, $U^{(n)}|_{n=1}^{N} \in \mathbb{R}_+^{I_n\times J_n} \geq 0$ is a set of factor matrices, $\underline{E}$ is the residual

tensor. Equivalently, NTD model can be written in matrix notation by use of Kronecker product as $X_{(n)} = U^{(n)} G_{(n)} U^{\otimes -n T} + E_{(n)}$, where $U^{\otimes -n} = U^{(N)} \otimes \cdots \otimes U^{(n+1)} \otimes U^{(n-1)} \otimes \cdots \otimes U^{(1)}$. As described in [1,9,12], the ALS update rules for factor matrices $U^{(n)}|_{n=1}^{N}$ and core tensor $\underline{G}$ are given by

$$U^{(n)} \leftarrow \left[X_{(n)} U^{\otimes -n} G_{(n)}^T \left(G_{(n)} (U^T U)^{\otimes -n} G_{(n)}^T \right) \right]_+ \tag{5}$$

$$\underline{G} \leftarrow \underline{G} \circledast \left[\left(\underline{X} \prod_{n=1}^{N} \times_n U^{(n)T} \right) \oslash \left(\underline{G} \prod_{n=1}^{N} \times_n U^{(n)T} U^{(n)} \right) \right] \tag{6}$$

The algorithm for ALS NTD is described in Algorithm 2.

Algorithm 2. ALS Algorithm for Nonnegative Tucker Decomposition

Data: Given data tensor $\underline{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N} \geq 0$, the component number $J_n|_{n=1}^{N}$.
Result: The estimated matrices $U^{(n)}|_{n=1}^{N}$, core tensor $\underline{G}$.

```
Initialization: Randomly nonnegative U^(n), G, normalize all U^(n)|_{n=1}^N;
repeat
    for n = 1 : N do
        U^(n) ← [X_(n) U^{⊗-n} G_(n)^T (G_(n) (U^T U)^{⊗-n} G_(n)^T)]_+
    G ← G ⊛ [(X ∏_{n=1}^N ×_n U^(n)T) ⊘ (G ∏_{n=1}^N ×_n U^(n)T U^(n))]
until convergence criterion is reached ;
```

2.3 Algorithm for K-CNTD

NTD model is a useful method for higher order data analysis. While the potential dependencies across the columns of factor matrices is ignored which is usually explored in cocktail party problem when analyzing speech signals.

In order to investigate the repeating patterns that span multiple columns of factor matrices, we extend NTD model into convolutive case according to convolutive NMF and NTD model. The convolutive NTD model in one mode is given by

$$\begin{aligned}
\underline{X} &= \underline{G} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)} + \underline{E} \\
&= \underline{G} \times_1 \left(\sum_{l=0}^{L-1} \overset{l\downarrow}{H}{}^{(1)} W_l^{(1)} \right) \times_2 U^{(2)} \cdots \times_N U^{(N)} + \underline{E} \\
&= \sum_{l=0}^{L-1} \left(\underline{G} \times_1 W_l^{(1)} \right) \times_1 \overset{l\downarrow}{H}{}^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)} + \underline{E} \\
&= \sum_{l=0}^{L-1} \underline{G}_l \times_1 \overset{l\downarrow}{H}{}^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)} + \underline{E}
\end{aligned} \tag{7}$$

Figure 1 illustrates the basic idea of convolutive NTD in one mode. More generally, we can extend equation (7) into convolutive case in K modes (K-CNTD) as

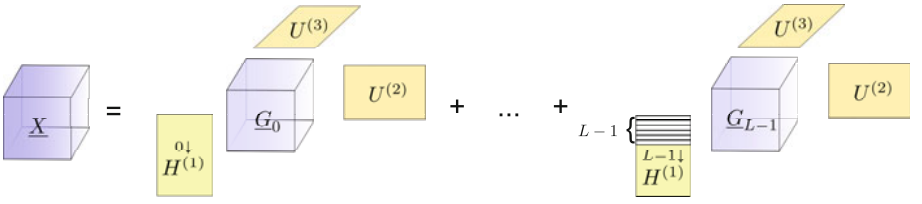


Fig. 1. Convolutive NTD in one mode

$$\begin{aligned}
\underline{X} &= \underline{G} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)} + \underline{E} \\
&= \underline{G} \times_1 \left(\sum_{l_1=0}^{L_1-1} \overset{l_1\downarrow}{H^{(1)}} W_{l_1}^{(1)} \right) \cdots \times_K \left(\sum_{l_K=0}^{L_K-1} \overset{l_K\downarrow}{H^{(K)}} W_{l_K}^{(K)} \right) \times_{K+1} U^{(K+1)} \cdots \times_N U^{(N)} + \underline{E} \\
&= \sum_{l_1=0}^{L_1-1} \cdots \sum_{l_K=0}^{L_K-1} \left(\underline{G} \prod_{k=1}^{K} \times_k W_{l_k}^{(k)} \right) \times_1 \overset{l_1\downarrow}{H^{(1)}} \cdots \times_K \overset{l_K\downarrow}{H^{(K)}} \times_{K+1} U^{(K+1)} \cdots \times_N U^{(N)} + \underline{E} \\
&= \sum_{l_1=0}^{L_1-1} \cdots \sum_{l_K=0}^{L_K-1} \underline{G}_{l_1 \cdots l_K} \times_1 \overset{l_1\downarrow}{H^{(1)}} \cdots \times_K \overset{l_K\downarrow}{H^{(K)}} \times_{K+1} U^{(K+1)} \cdots \times_N U^{(N)} + \underline{E} \qquad (8)
\end{aligned}$$

where the core tensors $\underline{G}_{l_1 \cdots l_K} = \underline{G} \prod_{k=1}^{K} \times_k W_{l_k}^{(k)}$, $l_k = 1, \cdots, L_k$, $k = 1, \cdots, K$. In this model, data tensor is decomposed into a set of core tensors $\underline{G}_{l_1 \cdots l_K}$, shifted factor matrices $H^{(k)}|_{k=1}^{K}$ and $U^{(n)}|_{n=K+1}^{N}$.

According to equation (8), we give a two stage algorithm for K-CNTD based on the ALS NTD [1,12] and ALS convolutive NMF with sparse constraint algorithm. The algorithm is described in Algorithm 3.

3 Simulation

3.1 Synthetic Data

A synthetic data tensor with three modes is generated to test the performance of K-CNTD algorithm. We use $S_1 \in \mathbb{R}^{2\times 1000}$ and $S_2 \in \mathbb{R}^{2\times 1000}$ as sources signal to generate convolutive mixture X_1 and X_2 respectively. Several samples of S_1 and S_2 are shown in Figure 2. The convolutive mixture $X_k = \sum_{l_k=0}^{L_k-1} A_{l_k}^{k} \overset{l_k\rightarrow}{S}_k$, $k = 1, 2$, where $A_{l_k}^{k}|_{l_k=0}^{L_k-1}$ are the mixture matrices.

We use X_1, X_2, $X_3 \in \mathbb{R}^{2\times 2}$ and $\underline{G} \in \mathbb{R}^{2\times 2\times 2}$ to generate a tensor $\underline{X}_{test} \in \mathbb{R}^{1000\times 1000\times 2}$ which can be seen as a mixture procedure in tensor structure by factor matrix X_3 and core tensor $\underline{G}$, i.e.

$$\underline{X}_{test} = \underline{G} \times_1 X_1 \times_2 X_2 \times_3 X_3 \qquad (9)$$

Algorithm 3. Algorithm for K-CNTD

Data: Given data tensor $\underline{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N} \geq 0$, the components number $J_n|_{n=1}^N$ for NTD, the convolutive length L_k, the components number T_k for CNMF, $(k = 1, \cdots, K)$.

Result: The estimated components $W_{l_k}^{(k)}|_{l_k=0,k=1}^{L_k,K}$, $H^{(k)}|_{k=1}^K$, $U^{(n)}|_{n=K+1}^N$, $\underline{G}_{l_1,\cdots,l_K}$.

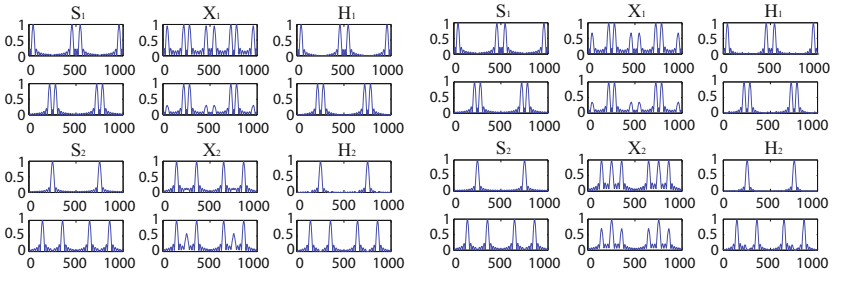
Initialization: Set $U^{(n)}|_{n=1}^N, \underline{G}$ randomly;

$[U^{(n)}|_{n=1}^N, \underline{G}] = \text{als-ntd}(\underline{X}, J_n|_{n=1}^N, U^{(n)}|_{n=1}^N, \underline{G})$; % Algorithm 2

for $k = 1 : K$ **do**

  $[W_{l_k}^{(k)}|_{l_k=0}^{L_k}, H^{(k)}] = \text{als-cnmf}(U^{(k)}, L_k, T_k)$; % Algorithm 1

$\underline{G}_{l_1,\cdots,l_K} = \sum_{l_1} \cdots \sum_{l_k} \underline{G} \times_1 W_{l_1}^{(1)} \cdots \times_N W_{l_K}^{(K)}$



(a) L=2 (b) L=4

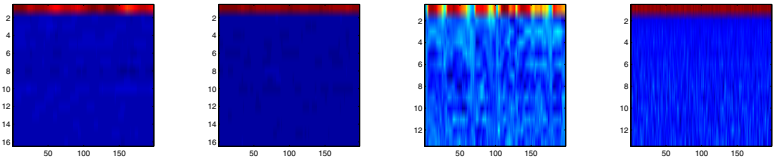
Fig. 2. Estimated results with convolution length $L = 2, 4$

We employ K-CNTD to recover the sources components and the estimated components are denoted as $H_k|_{k=1}^2$. Figure 2 gives the estimated signal with the convolution length $L = 2, 4$. From results, K-CNTD algorithm can recover the original signal from the tensor mixture.

3.2 Speaker Recognition in Noisy Condition

In this experiment K-CNTD algorithm is applied to feature extraction of speaker recognition in noisy conditions. Grid corpus (speech of 34 persons) mixed with different noise is used to test the recognition performance. We employ the cortical-based feature extraction framework described in [11] with 4-order tensor structure (time, frequency, scale and direction) and K-CNTD algorithm to extract the shift-invariant speech features in time-frequency domain. The sampling rate of speech signal is 8kHz. In order to estimate the speaker model and test the efficiency of our method, we use 1700 sentences (50 sentences each speaker) as training data and 2040 sentences (60 sentences each speaker) are selected for testing. We use the basis function $H^{(2)}$ in frequency mode to project the cortical representation into feature subspace and obtain the spare tensor features.

The testing samples in noisy condition are generated by mixing with Babble, Destroyerops, F16, Factory, Pink in SNR intensities of -5dB, 0dB, 5dB and 10dB respectively. For the final feature set, 16 cepstral coefficients were extracted and used for speaker modeling. GMM was used to build the recognizer with 64 gaussian mixtures.



(a) (b) (c) (d)

Fig. 3. (a)Clean feature extracted by K-CNTD.(b) Feature extracted by K-CNTD in 5dB condition with pink noise. (c) Clean MFCC. (d) MFCC in 5dB condition with pink noise.

For comparison, we test the performance of NTD, Spectral Substraction(SS) and MFCC. NTD is used to learning the basis functions and the feature extraction procedure is similar to the framework in [11]. Figure 3 gives the DCT feature comparison between MFCC and features extracted by K-CNTD in clean and 5dB conditions. The degradation of MFCC is evident. Comparison with the clean condition, the shift-invariant features extracted by K-CNTD maintain the useful information and provide robust and natural representation for speaker modeling. Also, the sparse constraint can make the feature robust because the energy of clean signal is concentrated on a few components only, while the energy of noises spread on all the components. Figure 4 presents the recognition accuracy in five noisy conditions averaged over SNRs between -5 to 10dB, and the overall average accuracy across all the conditions. The results suggest that our proposed K-CNTD algorithm can give a better average recognition result than NTD algorithm and traditional feature extraction methods.

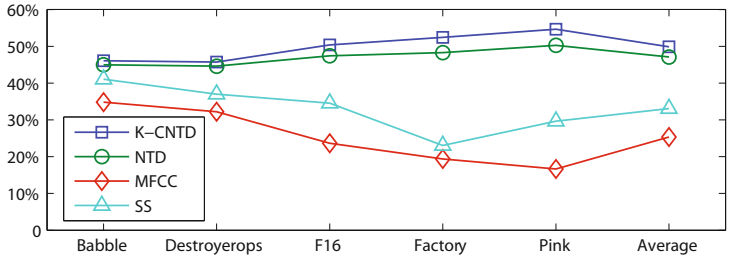


Fig. 4. Average speaker recognition accuracy in different condition

4 Conclusion

In this paper, we presented a two stage algorithm for the K-mode convolutive nonnegative tucker decomposition model. This model is an extension of nonnegative tucker decomposition and can preserve the intrinsic information in the natural structure of data. The shift-invariant feature in tensor structure can be extracted by the K-CNTD algorithm using the cortical representation. The experimental results indicate that our proposed algorithm is effective and robust for data exploratory, especially in the case of convolutive model.

References

1. Cichocki, A., Zdunek, R., Phan, A.H.: Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. John Wiley and Sons (2009)
2. Carroll, J.D., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition. Psychometrika 35, 283–319 (1970)
3. Kroonenberg, P.M., De Leeuw, J.: Principal component analysis of three-mode data by means of alternating least squares algorithms. Psychometrika 45, 69–97 (1980)
4. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM Journal on Matrix Analysis and Applications 21, 1253–1278 (2000)
5. Harshman, R.A., Hong, S., Lundy, M.E.: Shifted factor analysis Part I: Models and properties. Journal of Chemometrics 17, 363–378 (2003)
6. Mørup, M., Hansen, L.K., Arnfred, S.M., Lim, L.H., Madsen, K.H.: Shift-invariant multilinear decomposition of neuroimaging data. NeuroImage 42, 1439–1450 (2008)
7. Mørup, M.: Applications of tensor (multiway array) factorizations and decompositions in data mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1, 24–40 (2011)
8. Smaragdis, P.: Convolutive speech bases and their application to supervised speech separation. IEEE Transactions on Audio, Speech, and Language Processing 15, 1–12 (2007)
9. Cichocki, A., Phan, A.H.: Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE Transactions on Fundamentals of Electronics 92, 708–721 (2009)
10. Kim, Y.D., Choi, S.: Nonnegative tucker decomposition. In: CVPR 2007, pp. 1–8 (2007)
11. Wu, Q., Zhang, L.Q., Shi, G.C.: Robust feature extraction for speaker recognition based on constrained nonnegative tensor factorization. Journal of Computer Science and Technology 25, 783–792 (2010)
12. Phan, A.H., Cichocki, A.: Extended HALS algorithm for nonnegative Tucker decomposition and its applications for multiway analysis and classification. Neurocomputing 74, 1956–1969 (2011)

Making Image to Class Distance Comparable

Deyuan Zhang, Bingquan Liu, Chengjie Sun, and Xiaolong Wang

School of Computer Science and Technology, Harbin Institute of Technology, China
{dyzhang,liubq,cjsun,wangxl}@insun.hit.edu.cn
http://www.insun.hit.edu.cn

Abstract. Image classification is to classify the image into predefined image categories. The image to class distance(I2CD), with simple formulation, can tackle the intra-class variation and show the state of the art results in several datasets.

This paper focuses on the performance of I2CD on imbalanced training dataset which has not been catched much attention by I2CD researchers. Under the naive bayes assumption, when the dataset is imbalanced, I2CD is not comparable. We propose Random Sampling I2CD to tackle the imbalanced problem, and provide an efficient approximation method to reduce the test time complexity. Experimental results show that PRSI2CD outperforms the original I2CD in imbalanced setting.

Keywords: image classification, image to class distance, imbalanced dataset.

1 Introduction

Image classification is to classify the image into predefined image categories. Although it is easy for humans, it is proved to be a challenging task for computer vision systems. The difficulties mainly lie in the intra and inter class variation and different conditions such as lighting, scale, translation, occlusion and so on. In recent years, local patch based image descriptors such as SIFT[1] have high discriminative power between differnet image classes, and parametric methods such as Support Vector Machines[2], boosting[3], generative models[4] and so on have shown the state of the art performance.

Non parametric methods such as nearest neighbor, does not succeed in bag of words representation when there exist relative small training images. Boiman[5] study the problem, and showed that the poor performance is because of the information loss in the quantization step, and proposed Naive Bayes Nearest Neighbor(NBNN) image to class distance(I2CD) to avoid the feature quantization. They showed state of the art performance in several image datasets. Several extensions have been made by researchers to improve the performance of the image to class distance. Huang[6] used the image to class distance for face and human giat recognition. Behmo[7] pointed out when the number of image descriptors share a large variation, the image to class is not optimal. He proposed a max margin L_1 norm learning method to learn the optimal parameters.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 671–680, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Wang[8] observed that Boiman's method needs too much image descriptors and proposes a weighted image to class distance in a large margin framework. These extensions makes image to class distance more stable.

In this paper, we focus on the performance of the Image to Class Distance(I2CD) when applied to imbalanced training datasets, which have not been thoroughly studied before. The imbalanced dataset is common in real world. Under Naive Bayes framework, we show that when the training images are imbalanced, comparing the image to class distance to each category is not equal to minimize the mean classification error. Therefore I2CD is not comparable in imbalanced setting, and shows poor performance in practice. In order to solve the problem, we propose Random Sampling Image to Class Distance and an approximation method to reduce the time complexity of test performance.

The paper is structured as follows: in Section 2 we briefly review the Boiman's NBNN framework. In Section 3 we show that when the dataset become imbalanced, the image to class distance is not comparable, and propose Probabilistic Random Sampling Image to Class Distance(PRSI2CD) to tackle the problem. Section 4 learns the Weighted PRSI2CD using a max margin formulation. A quantitive evaluation of various methods on two image datasets is compared in Section 5. A conclusion is drawn in Section 6.

2 Review of Image to Class Distance Based on Naive Bayes Assumption

In this section we breifly review the basics of the Image to Class Distance method under Naive Bayes assumption. For simplicity, we consider binary classification problem. Given training images and the corresponding labels $x_i, y_i (i = [1, 2, ..., n], y_i \in [1, -1]$, each training image x_i is described as m_i bag of features $F_i = f_{i,1}, f_{i,2}, ..., f_{i,mi}$. For each category, the whole features in the same class compose the prototype of the category, and denoted as P_1,P_{-1} respectively.

Given a image Q, and the image feature is denoted as $F_Q = f_{Q,1}, f_{Q,2}, ..., f_{Q,mQ}$. When the prior of each class is uniform, a maximum-a-posteriori(MAP) classifier minimizes the mean classification error

$$C^* = \arg\max_C p(C|Q) = \arg\max_C p(Q|C) \tag{1}$$

Assume each image feature $f_{Q,i}$ in the image Q is independent to each other, the probability can be formulated as:

$$p(Q|C) = p(f_{Q,1}, ..., f_{Q,mQ}|C) = \prod_{i=1}^{mQ} p(f_{Q,i}|C) \tag{2}$$

The key is computing the probability density. Parzen Window[9] is often used to estimate the density, and Gaussian kernel function estimator is the most widely used:

$$p(f_{Q,i}|C) = \frac{1}{L_C}\sum_{j=1}^{L_C}\frac{1}{\sqrt{2\pi}\sigma_C}exp(-\frac{||f_{Q,i}-f_j||^2}{2\sigma_C^2}) \tag{3}$$

Where L_C denotes the total number of prototype P_C, and σ_C is the kernel width of category C. In theory, when descriptors grow to infinity, the kernel width reduces correspondingly, and equation 3 reaches the optimal solution of density. When the kernel width is little enough, the probability is dominated by the least distance of the image features in proto. Therefore when L_C is large enough, we can use the kernel density of nearest neighbor to approximate the probability:

$$p(f_{Q,i}|C) = \frac{1}{\sqrt{2\pi}\sigma_C L_C}exp(-\frac{||f_{Q,i}-NN_C(f_{Q,i})||^2}{2\sigma_C^2}) \tag{4}$$

When the proto number L_C of each class and the kernel width σ_C are equal, the probability can be reduced to calculate the following equation:

$$\log p(Q|C) = \sum_{i=1}^{mQ}\log p(f_{Q,i}|C) \propto -\sum_{i=1}^{mQ} \parallel f_{Q,i} - NN_C(f_{Q,i}) \parallel^2 \tag{5}$$

Define Image to Class Distance as follows:

$$Dist(Q,C) = \sum_{i=1}^{mQ} \parallel f_{Q,i} - NN_C(f_{Q,i}) \parallel^2 \tag{6}$$

Maximize the probability of image Q to class C is equal to minimize the Image to Class Distance.

3 Probabilistic Random Sampling Image to Class Distance (PRSI2CD)

The formulation of I2CD assumes using the same gaussian kernel width and same number of prototypes in each class. When the image dataset is imbalanced, both the kernel width and number of prototypes change accordingly. The number of prototypes can be easily obtained, and it is easy to incorporate into the original formulation. How to determine the kernel width can not be solved. Behmo[7] proposed a formulation to learn the optimal kernel width when the number of prototypes is not the same, but the formulation does not consider the influence of the imbalanced training images. Estimating the exact kernel width explicitly is hard in the imbalanced setting. Firstly, the number of image features are often large, resulting that the time complexity of estimating the kernel width is high. Secondly, which criterion is chosen for optimizing the optimal kernel width remains unclear in the imbalanced setting. Therefore we do not estimate the kernel width explicitly. We take another approach: for convenient suppose

$L_1 < L_{-1}$, we randomly draw L_1 prototype from P_{-1}, and repeat n epoches in which the average distance of image to class is computed. It is comparable under the naive bayes assumption. We call the distance Random Sampling Image to Class Distance(RSI2CD).

$$Dist(Q, C, L_1) = \frac{1}{n}\sum_{j=1}^{n} Dist(Q, C_j) \quad C_j \subset P_C, |C_j|_0 = L_1 \tag{7}$$

Denote L_2 as the number of prototypes of class C, in this paper, $L_2 = L_1$ or L_{-1}. In theory, when we repeat more random sampling procedure, the RSI2CD is closer to its real value. More sampling procedures means the time complexity grows. In order to reduce the time complexity we propose the following approximation methods. The equation can be written as the average distance of each image descriptor:

$$Dist(Q, C, L_1) = \sum_{i=1}^{mQ} \frac{1}{n}\sum_{j=1}^{n} \parallel f_{Q,i} - NN_{Cj}(f_{Q,i}) \parallel^2 \quad C_j \subset P_C, |C_j|_0 = L_1 \tag{8}$$

If we can compute the average distance of each feature of image Q, RSI2CD can be computed easily using Equation 8. When n approaches infinity, we will get the expectation distance of $f_{Q,i}$. There exist $C_{L_2}^{L_1}$ combinations of sampling, defined as follows:

$$C_{L_2}^{L_1} = \frac{L_2 \times (L_2 - 1) \times ... \times (L_2 - L_1 + 1)}{L_1!} \tag{9}$$

Define $f_1, f_2, , f_K$ is the K nearest neighbor of the image feature $f_{Q,i}$. We will compute from combinations how many times f_t will be chosen as the nearest neighbor of $f_{Q,i}$. The f_t is chosen as the nearest neighbor of $f_{Q,i}$ means that $f_1, f_2, ..., f_{t-1}$ is not sampling from the L_2 protos, and f_t is chosen. Therefore there exists $C_{L_2-t}^{L_1-1}$ times f_t is chosen as the nearest neighbor of $f_{Q,i}$. Thus the probability of f_t is chosen as the nearest neighbor of $f_{Q,i}$ is:

$$p(f_t = NN_C(f_{Q,i})) = \frac{C_{L_2-t}^{L_1-1}}{C_{L_2}^{L_1}} \tag{10}$$

The probability of each feature can be calculated explicitly, thus the average distance can be computed precisely. In practice, each image descriptor have the probability of chosen as the nearest neighbor, and calculating the exact probability of each proto needs $2 * (L_2 - t)$ multiplications. In order to reduce the time complexity of the computation of exact average distance, we need to compute the approximate probability of each K nearest neighbor. Compare the probability of f_t and f_{t+1} chosen as the nearest neighbor, we can get:

$$\frac{p(f_t = NN_{Cj}(f_{Q,i}))}{p(f_{t+1} = NN_{Cj}(f_{Q,i}))} = \frac{C_{L_2-t}^{L_1-1}}{C_{L_2-t-1}^{L_1-1}} = \frac{L_2 - t}{L_2 - L_1 - t + 1} \tag{11}$$

When $t \ll L_1$, the ratio is approximately equal to $L_2/(L_2 - L_1)$. From equation 10 we can get the probability of f_1 is chosen as $f_{Q,i}$'s nearest neighbor is L_1/L_2. The probability $f_1, f_2, ..., f_K$ is a Geometric Sequence, thus the probability of top K proto chosen as the nearest neighbor is:

$$p(f_t = KNN_{Cj}(f_{Q,i})) = 1 - (\frac{L_2 - L_1}{L_2})^K \tag{12}$$

It is obvious that the probability is heavily long tail, thus we can approximate the average distance using top K nearest neighbor. In this paper, we set the threshold value to 0.9, so K is chosen as:

$$\max_K p(f_t = KNN_{Cj}(f_{Q,i})) = 1 - (\frac{L_2 - L_1}{L_2})^K \geq 0.9 \tag{13}$$

Resulting in K is calculated as:

$$K = \left\lceil \frac{ln0.1}{ln(\frac{L_2 - L_1}{L_2})} \right\rceil \tag{14}$$

Thus the approximate RSI2CD can be calculated using top K nearest neighbors:

$$Dist(Q, C, L_1) = \sum_{i=1}^{mQ} \sum_{t=1}^{K} p(f_t = NN_{Cj}(f_{Q,i})) \parallel f_{Q,i} - NN_{Cj}(f_{Q,i}) \parallel^2 \tag{15}$$

The approximate RSI2CD is called Probabilistic RSI2CD(PRSI2CD). Other threshold value can be chosen manually, in the experiment the threshold 0.9 performs well.

4 Weighted PRSI2CD

In order to represent the image to class distance conveniently, the PRSI2CD is transformed into vector form. Denote D_1, D_2 as the vectors for image Q with length $L_1 + L_{-1}$, and the value of each dimensionality is:

$$D_1(j) = \begin{cases} 0 & \text{if } j \geq L_1 \\ \sum_{i=1}^{mQ} p(f_j = NN_{C_1}(f_{Q,i})) \parallel f_{Q,i} - NN_{C_1}(f_{Q,i}) \parallel^2 & \text{if } j < L_1 \end{cases} \tag{16}$$

$$D_2(j) = \begin{cases} \sum_{i=1}^{mQ} p(f_j = NN_{C_{-1}}(f_{Q,i})) \parallel f_{Q,i} - NN_{C_{-1}}(f_{Q,i}) \parallel^2 & \text{if } j \geq L_1 \\ 0 & \text{if } j < L_1 \end{cases} \tag{17}$$

Thus $e^T D_1, e^T D_2$ represents the PRSI2C Distance of image Q to class C_1 and C_{-1} respectively, where e is the row vector whose each element is 1. Inspired by Wang's[8] formulation, we extend the PRSI2CD to the Weighted PRSI2CD by

replacing e with W whose elements are not equal with each other and not less than 0. We set different penalties for different class. For binary classification, the optimization is formulated as follows:

$$\begin{aligned} \min & \frac{1}{2} \parallel W - W_0 \parallel^2 + C_1 \sum_{y_i(y_i=1)} \xi_i + C_2 \sum_{y_i(y_i=-1)} \xi_i \\ & s.t. \\ \forall i \quad & y_i W^T (D_2(x_i) - D_1(x_i)) \geq 1 - \xi_i \\ & \xi_i \geq 0 \\ \forall k \quad & W(k) \geq 0 \end{aligned} \tag{18}$$

Where W_0 is a prior weight vector, in the paper, all the elements of W_0 are set to 1. The penalty parameters C_1 and C_2 are set to different value on different class to tackle the imbalanced dataset. For simplicity we set $D_i = D_2(x_i) - D_1(x_i)$. As the method in [8], we solve the optimization problem in the dual form:

$$W = W_0 + \sum_{i=1}^{n} \alpha_i y_i D_i + \mu \tag{19}$$

We iteratively update the value α_i and μ using the following update rules:

$$\alpha_i \leftarrow [\frac{1 - \sum_{j \neq i} \alpha_j y_j < D_j \cdot D_i > - < (\mu + W_0) \cdot D_i >}{\parallel D_i \parallel^2}]_{(0, C_{y_i})} \tag{20}$$

$$\mu \leftarrow max\{0, -\sum_{i=1}^{n} \alpha_i y_i D_i - W_0\} \tag{21}$$

The difference with Wang's formulation is that we set different penalty parameters on different class, which is often used in large margin framework to tackle the imbalanced training dataset.

5 Experiments

In order to test our method can tackle imbalanced datasets, we compare the NBNN[5] method and I2CD[8] as a baseline. We test these methods on two datasets, mainly in imbalanced setups. When the datasets become balanced, our method is the same with pervious methods. In order to make a comprehensive comparison, we use four evaluation criterions. Mean Accuracy for each class is used as an evaluation measure. Binary classification problem can be treated as detection problem, and Receiver Operating Characteristic[10](ROC) curve is often used as the measure of detection performance. Based on ROC curve, the Area Under Curve(AUC) and Equal Error Rate(EER) are used for the evaluation measure. In recent study[11], Average Precision (AP) showed better

characteristics than the EER and AUC, and we include AP as a evaluation criterion. We compare the performance of these methods on Graz-01 datasets and Caltech-101 databases.

Before we compare the performance on different datasets, we first introduce the experimental setup. Before the feature extraction images whose width or heights are larger than 300 pixels are scaled to 300 pixels with preserving the aspect ratio of the image, and SIFT [1] descriptors are computed on 16×16 pixel patches over a grid with spacing of 8 pixels. For the weighted distance the penalty parameter C_1 of positive training images is set to 1, and C_{-1} is set to L_1/L_{-1}.

5.1 Graz-01 Dataset

The dataset contains 373 images of the category "bike", 460 images of the category "person", and 273 "background" images. The images are highly complex with high intra-class variability in scale, viewpoint, color, location and illumination. There is much background clutter in the image. These characteristics make the image classification algorithms difficult for recognizing the categories. The categorization task is Class vs Others. We randomly draw 100 images from the category (bike or person) as positive training images and 100 images as negative images (of which 50 images are sampled from the counter category and 50 other images are sampled from the "background" category). Similar strategy is adopted to obtain test set. In order to test our method for dealing with imbalanced database, we change the ratio of positive images to negative images for training the classifier. We experiment the ratio from 1:5 to 1:1, that is, we choose the 20, 25, 34, 50 and 100 positive images and 100 negative images for training.

The performances on "bike" and "person" are compared in Fig. 1 and Fig. 2 respectively. As Fig. 1 and Fig. 2 show, PRSI2C and W-PRSI2C outperform the baseline methods when the dataset is imbalanced. Compared to the "bike" category, classifying the "person" category is more challenging. "Person" category shares more intra-class variability: people have different poses, and the background is more sophisticated. This results in the difficulty of recognizing person category. When the dataset is imbalanced, the performance our method did not degrade much. The performance of using 25 positive training images is comparable with that of using 100 positive training images. One exception is on the "bike" category using 50 positive training images.

5.2 Caltech101 Dataset

The Caltech-101 dataset consists of 101 object categories and an additional class of background images. Although the database has less intra-class variability, it is still a challenging database because of high inter-class variability and the large number of categories.

Binary classification scheme is adopted in the paper. We randomly choose 15 images of each category for positive images, and select N images from each category for negative images. We compare the performance of N=1,2,3. That is 101, 202, 303 images are selected to form negative images respectively. Similar

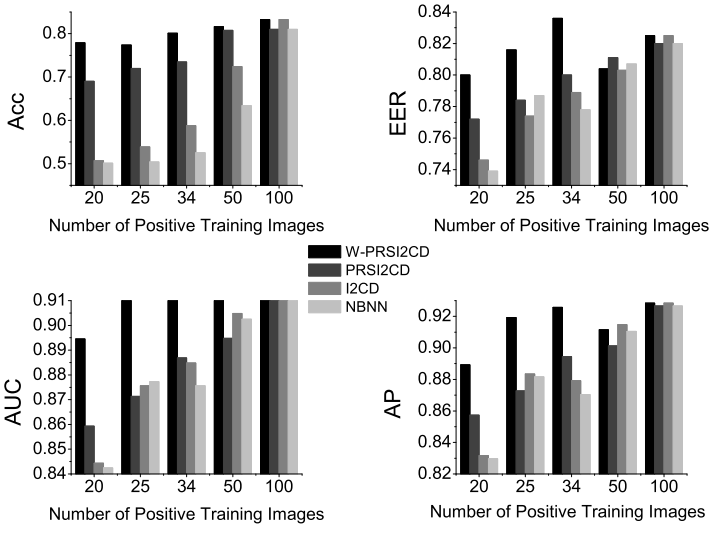


Fig. 1. The performance(Acc, EER, AUC, AP) on Graz-01 "bike" category

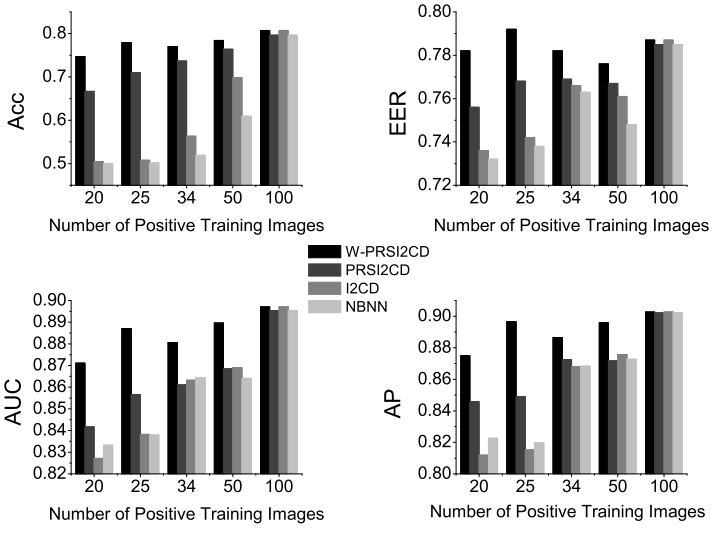


Fig. 2. The performance(Acc, EER, AUC, AP) on Graz-01 "person" category

strategies are used to form test images. We randomly experiment 10 runs. Because there exist 101 binary classifications, we can not report the performance of each class in detail. The Mean of evaluation measure of the 101 class is reported in Fig. 3. From Fig. 3 we can see that the performance of our distance is more superior to the baseline methods in each evaluation measure. The Accuracy of baseline methods is always about 0.5, because most of positive images are classified as negative ones. Our method can tackle the imbalanced dataset. Another observation is that although the I2CD outperforms NBNN, the performance of I2CD has a larger degradation than NBNN. That is I2CD is more sensitive to the imbalanced datasets. This confirms that under the wrong condition, weighted learning perhaps gives more bad results.

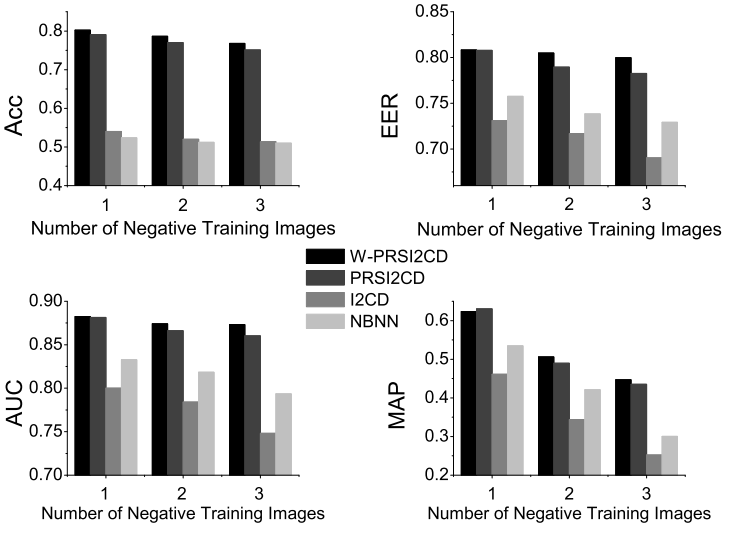


Fig. 3. The performance(Acc, EER, AUC, AP) on Caltech101 dataset

In the setting, the EER and AUC always keep high performance (about 80%). This is because according to the definition of AUC and EER, when some images from the positive class are classified by the classifier, the EER and AUC are always high. There exist some images that are easy to classify and others are not, the easy ones can be classified correctly by each classifier, resulting the high performance of NBNN and I2CD. Our method outperforms the baseline methods stably.

6 Conclusions

This paper focus on the image to class distance on imbalanced datasets. We see our contribution as follows: (1)From naive bayes assumption, we illustrated that minimize the image to class distance is not equal to maximize the Maximum-A-Posteriori, and is not comparable when the dataset is imbalanced. (2)We

propose Random Sampling Image to Class Distance to tackle the problem, and an approximate method(PRSI2C) for efficient calculation. (3)We propose a large margin framework to learn the optimal PRSI2C that considers the imbalanced issue. Imbalanced datasets are common in real world, and this is the first step to make the Image to Class Distance toward more sophisticated and challenging classification settings.

Acknowledgments. This work was funded in part by the National Natural Science Foundation of China (Grant No. 60973076) and the Special Fund Projects for Harbin Science and Technology Innovation Talents (2010RFXXG003).

References

1. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
2. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning 20(3), 273–297 (1995)
3. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic Object Recognition with Boosting. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(3), 416–431 (2006)
4. Li, F.F., Fergus, R., Perona, P.: One-shot Learning of Object Categories. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(4), 594–611 (2006)
5. Boiman, O., Shechtman, E., Irani, M.: In Defense of Nearest-Neighbor Based Image Classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE Press, New York (2008)
6. Huang, Y., Xu, D., Cham, T.J.: Face and Human Gait Recognition using Image-to-Class Distance. IEEE Transactions on Circuits and Systems for Video Technology 20(3), 431–438 (2010)
7. Behmo, R., Marcombes, P., Dalalyan, A., Prinet, V.: Towards Optimal Naive Bayes Nearest Neighbor. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 171–184. Springer, Heidelberg (2010)
8. Wang, Z., Hu, Y., Chia, L.-T.: Image-to-Class Distance Metric Learning for Image Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 706–719. Springer, Heidelberg (2010)
9. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. Wiley-Interscience, USA (2001)
10. Fawcett, T.: An Introduction to Roc Analysis. Pattern Recogn. Lett. 27, 861–874 (2006)
11. Nowak, S., Lukashevich, H., Dunker, P., Rüger, S.: Performance Measures for Multilabel Evaluation: a Case Study in the Area of Image Classification. In: Proceedings of the International Conference on Multimedia Information Retrieval, New York, USA, pp. 35–44 (2010)

Margin Preserving Projection for Image Set Based Face Recognition

Ke Fan, Wanquan Liu, Senjian An, and Xiaoming Chen

Department of Computing, Curtin University, WA 6102, Australia
{ke.fan,xiaoming.chen}@postgrad.curtin.edu.au,
{W.liu,S.An}@curtin.edu.au

Abstract. Face images are usually taken from different camera views with different expressions and illumination. Face recognition based on Image set is expected to achieve better performance than traditional single frame based methods, because this new framework can incorporate information about variations of individual's appearance and make a decision collectively. In this paper we propose a new dimensionality reduction method for image set based face recognition. In the proposed method, we transform each image set into a convex hull and use support vector machine to compute margins between each pair sets. Then we use PCA to do dimension reduction with an aim to preserve those margins. Finally we do classification using a distance based on convex hull in low dimension feature space. Experiments with benchmark face video databases validate the proposed approach.

Keywords: Face recognition, Dimensionality reduction, Support vector machine, Image set match, Convex hull distance.

1 Introduction

In traditional face recognition, a classifier is trained from training samples and an inquiry image identity is recognized based on only one testing image. Under controlled conditions, conventional methods can achieve satisfactory performance. However, most of these approaches can not exploit facial variations very efficiently, since they don't use the variation information collectively. The performance is generally unsatisfactory in uncontrolled or semi-controlled conditions, for example, for images captured in surveillance system and web camera. Nowadays, it is easy to obtain large quantity of images for both training and testing. Theoretically, a set of images for the same individual should provide more variation information in pose, illumination and expression. In this situation, face recognition problem can be formulated as follows: Given a set of query images for one unknown subject, we need to design a classifier based on the training image sets and use it to find the label information for the query image set. This is called Face Recognition based on Image Set (FRIS).

Initially, some researches attempt to implement frame based methods for FRIS problem. They apply frame based techniques to all or selected frames from face

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 681–689, 2011.
© Springer-Verlag Berlin Heidelberg 2011

sequences, and then obtain corresponding results using majority voting or other decision level fusion algorithms [1]. This strategy ignores some important information for the correlation in the testing set. Experiments show that image set based methods outperform those single frame based ones [2,3] with direct applications on FRIS.

In order to solve FRIS problem, several methods have been proposed recently. Some methods purely concern how to measure the similarity of two sets [3,4,5]. In this case, each image set is generally considered as a linear subspace [4,6] or a nonlinear manifold [2,3,7]. A computationally efficient way of computing the similarity between two linear spaces is to calculate their canonical correlation, which is defined as cosines of principal angles [6,8]. Motivated by Mutual Subspace Method (MSM) [4], some dimensionality reduction methods for image sets are proposed [6,9,10].

Recently a state-of-the-art work [5] characterizes each image set as a convex region and proposes a new metric for image sets, which is defined as the minimum distance between points in convex sets. The experiments in [5] show that this approach can achieve better performance than previous works [2,4,7]. Technically, [5] mainly concerns how to measure the similarity of two image sets, but pay less attention to discriminant learning with such similarity. Further, the classification in [5] is done in original dataset without dimensionality reduction. In this paper, a novel linear dimensionality reduction algorithm for FRIS is proposed. We intend to compute the convex hull distance by definition. But when two image sets are inseparable, direct computing is not suitable. In this case, SVM is implemented to handle this problem. The margin obtained from SVM is used to approximate the convex hull distance. Then using PCA on directions derived from SVM, one can find a subspace spanned by the dominant directions. Finally, classification is implemented in the reduced feature space based on the convex hull distance.

The rest of this paper is organized as follows: Some relevant methods are briefly discussed in Section 2. The new algorithm is presented in Section 3. The experimental results and discussions are presented in Section 4. Conclusions are drawn in Section 5.

2 Preliminaries

Let N_c be the number of classes and n_c $(c = 1, \ldots, N_c)$ be the number of data samples belonging to the cth class $(\sum_c n_c = N)$. The input data set $X_c = \{x_{ck} \in R^d, k = 1, \ldots, n_c\}$ is sampled from the cth class. The proposed method aims to perform dimensionality reduction from the input space data points $X \in R^{d \times N}$ to a lower dimensional feature space $Z = [z_1, \ldots, z_N] \in R^{m \times N} (m \ll d)$ for FRIS.

2.1 Convex Model

An intuitive idea for FRIS, is to approximate each image set with a convex model [5]. For an unknown set we try to find its class label by using distance

between convex models of testing and training image sets. The label is assigned to the training set which is closest to the testing set.

There are two major convex models, affine hull and convex hull. In original pixel space, the dimension of affine hulls is less than d, and this necessarily holds for $n_c \ll d$. The affine space is a subset of R^d. But in low-dimension feature space, the affine hulls have dimension nearly or exactly same as the feature space dimension m. The comparability for two sets will be lost, because the affine spaces of two image sets may easily overlap even though they are separable. The restricted linear combination coefficients of a convex hull make a tighter convex model and reduce the chance of overlap in low dimension space. In this paper, we focus on the convex hull model where each image set is modeled by:

$$H_c = \{x = \sum_{k=1}^{N_c} \alpha_k x_{ck} | \sum_{k=1}^{N_c} \alpha_k = 1, \alpha_k \geq 0\} \tag{1}$$

Suppose we have two image sets X_i and X_j. The distance between them is defined as the minimum distance between any point in convex hull H_i and any point in convex hull H_j:

$$d^c(X_i, X_j) = D(H_i, H_j) = \min_{x \in H_i, y \in H_j} \|x - y\| \tag{2}$$

where H_i and H_j include X_i and X_j respectively. In fact, the distance can be computed by solving the following optimization problem:

$$\begin{aligned} (\alpha^*, \beta^*) &= \arg\min_{\alpha,\beta} \|X_i\alpha - X_j\beta\|^2 \\ s.t. \quad & \sum_{k=1}^{n_i} \alpha_k = \sum_{k'=1}^{n_j} \beta_{k'} = 1, \quad \alpha_k, \beta_{k'} \geq 0 \end{aligned} \tag{3}$$

For convenience, we denote the distance by $d^c(X_i, X_j) = \|X_i\alpha^* - X_j\beta^*\|$. The constraints of (3) are not standard and sometimes the solution is not unique. The possibility of solving (3) is under the assumption that two image sets are linearly separable. If two image sets are not linearly sparable, the defined distance will be zero. In this case, SVM will be used to find support vectors, which can be used to approximate the similarity between two image sets.

2.2 SVM Approximation

Suppose we have some training data x_k with corresponding class label $y_k \in \{-1, 1\}$. SVM aims to find a decision hyperplane, which can be described by $w^T x + b = 0$, through maximizing margin. We can see that the concept of the *margin* in SVM, "which is defined to be the smallest distance between the decision boundary and any of the samples" [12], is closely related to the convex hull distance. The margin and convex hull distance are based on exactly the same boundary points, if the two sets are separable. However, SVM is still applicable

when two sets have overlap. Usually, we solve the following SVM optimization problem

$$\arg\min_{w} \quad \frac{1}{2}\|w\|^2 + C\sum_{k}\xi_k \qquad s.t. \quad y_k(w^T x_k + b) \geq 1 - \xi_k, \quad \xi_k \geq 0 \tag{4}$$

to find the vector w perpendicular to the decision hyperplane. w is the direction of margin, and the length of margin is given by $2/\|w\|$. In (4) minimizing $\|w\|$ is equivalent to maximizing the margin. When training data are projected along the direction w, the margin in the reduced subspace remains the same as in the original space. Actually, (4) can be rewritten as follows:

$$\arg\max_{w_{ij}} d^c(w_{ij}^T X_i, w_{ij}^T X_j) \tag{5}$$

After problem (4) is solved, w can be found and the distance between two image sets can be approximated by $d^c(X_i, X_j) = \frac{2}{\|w_{ij}\|}$.

3 Margin Preserving Projection

In this section, we introduce a new dimensionality reduction approach for FRIS. The problem is described as follows. Given training sets $[X_1, \cdots, X_c, \cdots, X_C]$, we intend to find an optimal projection A, which maps all sets to a bunch of low dimensional sets Y_c, with $Y_c = A^T X_c$. We expect this projection to keep sufficient discriminant information through preserving margins between any two sets in lower dimensional spaces.

3.1 The Proposed Algorithm

The proposed algorithm is named Margin Preserve Projection (MPP) and it includes the following three major steps:

1. **Finding the maximum margin directions:** Let w_{ij} be the maximum margin direction between sets X_i and X_j. We can obtain w_{ij} by solving SVM (4). (Only for $i < j$)
2. **Choosing the weights:** In order to preserve the local structure between two image sets, we calculate the relationship between them. This is motivated by Locality Preserving Projections (LPP) [17]. A possible choice of S_{ij} is $S_{ij} = \exp(-\frac{d^c(X_i,X_j)^2}{\sigma^2})$ where σ is a suitable constant. $d^c()$ is computed using the margin of SVM. One simple way of selecting parameter σ is $\sigma = mean(d^c(X_i, X_j))$.
3. **Solving eigenvector problem:** Compute the eigenvectors and eigenvalues of the scatter matrix: $P = \sum \left(\frac{w_{ij}}{\|w_{ij}\|}\right)\left(\frac{w_{ij}}{\|w_{ij}\|}\right)^T S_{ij}$, where $\frac{w_{ij}}{\|w_{ij}\|}$ is a normalized direction of w_{ij}. Let the column vectors $a_1, \cdots, a_m$ be the eigenvector of P, ordered according to their corresponding eigenvalues $\lambda_1 > \cdots > \lambda_m$. The projection matrix is $A = [a_1, \cdots, a_m]$ with size $d \times m$, and dimensionality reduction can be easily implemented by $Y_c = A^T X_c$ where the dimension of Y_c is m.

3.2 Intuition of MPP

The principal idea of this approach is underlying Principal Component Analysis (PCA) [13]. The aim of PCA is to project the data onto a low dimensional space which maximizes the variance of the projected data. The objective function of a general PCA is stated as below:

$$\arg\max_{A} \quad \sum_{ij} d(A^T x_i, A^T x_j)^2 \qquad s.t \quad A^T A = I \tag{6}$$

where $d()$ is the distance between two points, which is generally chosen to be the Euclidean distance. Following this idea, we expect to find a projection matrix A which can maximize the convex hull distances among image sets. The objective function can be intuitively modified as follows:

$$\arg\max_{A} \quad \sum_{i<j \quad i,j\in\{1,\cdots,C\}} d^c(A^T X_i, A^T X_j)^2 S_{ij} \qquad s.t \quad A^T A = I \tag{7}$$

Here we weight each distance by S_{ij}. Since the closest two sets are the most difficult to classify, their nearest neighbors involve most important information for discrimination. According to the above analysis, we put larger weights on smaller distances.

Since A is unknown, we can hardly compute the distance $d^c()$ with variable A. In fact solving (7) directly is very difficult. Instead, we find A through the maximum-margin directions w_{ij} ($i < j$ and $i, j \in \{1, \cdots, N_c\}$), which are obtained by solving the SVM optimization problem (4).

Let W be the space spanned by the $\frac{N_c(N_c-1)}{2}$ direction vectors w_{ij}. The intrinsic dimension M_W of this space satisfies $0 \leq M_W \leq \frac{N_c(N_c-1)}{2}$. And the dimension of projection A is in the range $0 \leq M_A \leq M_W$.

Information contained in the subspace A may not be capable to maximize all the pairwise distance, therefore we have:

$$\max \sum d^c(A^T X_i, A^T X_j)^2 S_{ij} \leq \max \sum d^c(W^T X_i, W^T X_j)^2 S_{ij} \tag{8}$$

Let $\tilde{W} \in R^{M_A}$ be a subspace of W. $\tilde{W}$ can be defined as M_A dominating directions of w_{ij}. The training error

$$\tau = \max \sum d^c(W^T X_i, W^T X_j)^2 S_{ij} - \max \sum d^c(\tilde{W}^T X_i, \tilde{W}^T X_j)^2 S_{ij} \tag{9}$$

should be very small, when unimportant directions are ignored. In this case, $\tilde{W}$ can be seen as an approximation of the optimal projection A.

Base on above analysis, the approximation of optimum projection A can be found as eigenvectors corresponding to the largest M_A eigenvalues for the pairwise scatter matrix P. Choosing enough eigenvectors can guarantee that the margins are mainly preserved. Abandoning eigenvectors corresponding to small eigenvalues will reduce noise.

In general, we expand PCA for the case of image sets by replacing point-to-point distances with set-to-set convex hull distances. Our approach is to find a subspace spanned by dominant projection directions of all w_{ij}. This subspace provides enough information for a convex distance classifier. After projection, we expect improvements on classification performance and reduction on computational cost.

4 Experiments

We tested the proposed method on two benchmark databases: Honda/UCSD [14] and CMU MoBo [15]. These two sets contain several videos each recording one subject's movement. We use a Viola-Jones face detector [16] to find all facial images used for training and testing. Before experiments all detected images were histogram normalized to eliminate some lighting effects.

The Honda/UCSD Video Database was collected for video-based face recognition. It contains 62 video sequences (including videos with partial occlusion) of 20 different people. It divides into two subsets: 20 videos for training and the remaining 42 videos for testing. Each cropped facial image was normalized to 40×40 gray scale image. Figure 1a presents some images from this database that belong to the same subject. From each training and testing set in this database, we build a randomly selected corresponding subset which contains 50% quantity of images and do experiment on those subsets. The experiments are repeat for 10 times and we obtain the average performance.[1]

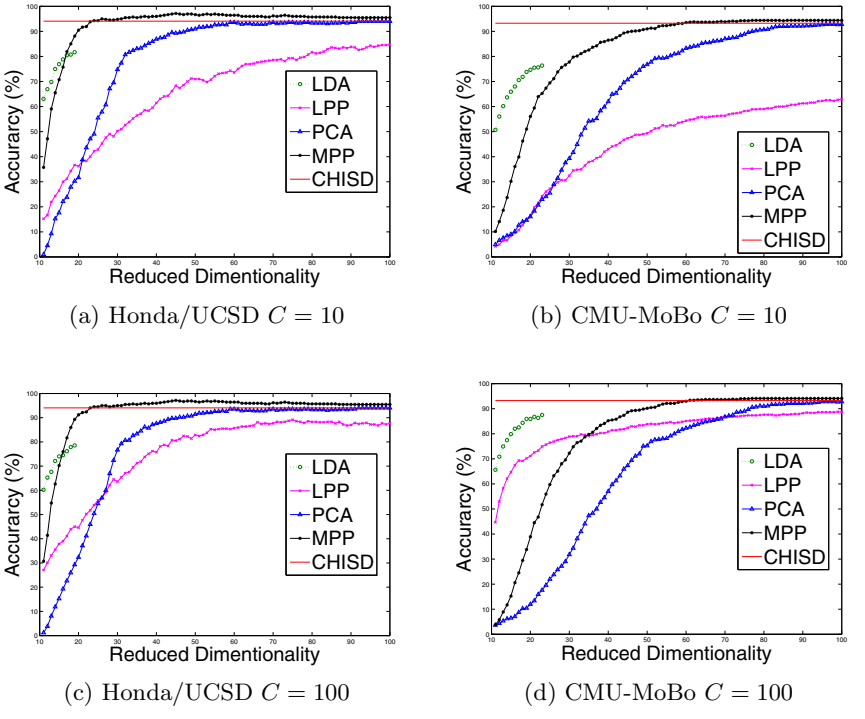


(a) Honda/UCSD (b) CMU MoBo

Fig. 1. Facial images detected from Honda/UCSD and CMU-MoBo database

The CMU MoBo database contains video of 24 individuals walking on a treadmill in an indoor environment. There are totally 96 sequences for 24 subjects, and each person has 4 sets. Each detected image was resized to 40×40 gray scale image. Figure 1b shows some examples of the detected faces from one subject. In this experiment, we randomly select one set of four from each subject for training and remaining 3 for testing. As before, all the experiments are repeated for 10 times and we report the average results.

[1] We did not use the setup for Honda/UCSD database in [5], because the number of testing sets is too small.



(a) Honda/UCSD $C = 10$ (b) CMU-MoBo $C = 10$

(c) Honda/UCSD $C = 100$ (d) CMU-MoBo $C = 100$

Fig. 2. Comparison of the averaged accuracy versus the reduced dimension of LDA, LPP, PCA and MPP on the Honda/UCSD and CMU-MoBo database

The methods compared here include: Manifold-Manifold Distance (MMD) [3], Convex Hull based Image Set Distance (CHISD) [5], Locality Preserving Projections (LPP) [17], Linear Discriminant Analysis (LDA) [11], Principal Component Analysis (PCA) [13] and the proposed method MPP. We tested CHISD and MMD in the original pixel feature space as baselines. For all other methods, we do dimensionality reduction first and then implement CHISD in these corresponding reduced feature spaces. For MMD, we use the same setup of parameters in [3]. For simplicity, we set $k = 10$ the number of neighbors in LPP. The penalty parameter C in SVM varies from 10 to 100 to explore its effects.

4.1 Experimental Results and Discussions

Figure 2 shows the averaged recognition accuracy of LDA, PCA, LPP and MPP under different reduced dimensions ($m = 11, \cdots, 100$). One exception is LDA which only can extract at most $N_c - 1$ meaningful dimensions. The best recognition rates and the averaged running time are shown in Table 1. Some interesting observations are provided on the performance of the evaluated algorithms.

Table 1. Comparison of the related algorithms to MPP on the Honda/UCSD and CMU-MoBo database. In accuracy, the first number is the highest recognition rates through different reduced dimensions; the following number is the corresponding dimension. The running time in seconds is the average time consumed on testing one set with the best dimension.

Algorithms	Honda/UCSD		CMU MoBo	
	accuracy	time(s)	accuracy	time(s)
MMD	79.76%	1.13	82.54%	182.20
CHISD	94.05%	6.27	93.23%	18.21
LDA+CHISD	80.00%,19	0.04	84.37%,20	0.38
LPP+CHISD	89.52%,75	0.14	82.82%,87	1.46
PCA+CHISD	94.05%,92	0.12	92.81%,94	2.74
MPP+CHISD	**97.14%,44**	0.10	**94.09%,76**	1.76

Firstly, for most methods, recognition rate increase consistently when the reduction dimensions increase. It can be seen that the two traditional methods - LDA and LPP yield poor performances. The performance of PCA is better than LDA and LPP, but not overtakes the baseline CHISD. Moreover, in a single frame based recognition problem, PCA can improve performance by preserving principal data variances, but here it is just similar to baseline CHISD. This result is different from some previous frame based experiments [17]. Though it may be not fair to compare them here, the experiments suggest that traditional dimensional reduction methods may be unsuitable for set based classification. This is due to the fact that PCA, LDA and LPP are performed on data points, but the final classifier is based on image sets. Secondly, the proposed MPP gives superior results than other methods with the best performance. Unlike other methods, the proposed MPP method preserves the margins, especially smaller ones which contain more discriminant information. By focusing on set based information, MPP method hence provides significant performance benefits. Notice that the best result does not come with highest dimension, which means that abandoned eigenvectors contain noises and have no help in recognition. Finally, CHISD, MPP and PCA are very stable when C changes. However, the performance of LDA and LPP are susceptible to variation of C. In fact, C represents the training error of SVM. Less sensitive to different C implies that these approaches have good generalization capability.

5 Conclusions

In this paper, we proposed a new linear dimensionality reduction algorithm called margin preserving projections. It is based on the metric of convex hulls for FRIS. The most interesting feature of this method is that it focuses on the relation between image sets rather than single images. This allows to retain more important information for set based classification problems. Experiments on face image databases show that the proposed method produces better recognition accuracy and less time consuming than some related algorithms.

References

1. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Computing Surveys 35(4), 399–458 (2003)
2. Hadid, A., Pietikainen, M.: From still image to video-based face recognition: an experimental analysis. In: Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 813–818 (2004)
3. Wang, R., Shan, S., Chen, X., Gao, W.: Manifold-manifold distance with application to face recognition based on image set. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2008)
4. Yamaguchi, O., Fukui, K., Maeda, K.i.: Face recognition using temporal image sequence. In: IEEE Conference on Automatic Face and Gesture Recognition, pp. 318–323. IEEE (1998)
5. Cevikalp, H., Triggs, B.: Face recognition based on image sets. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2567–2573. IEEE (2010)
6. Kim, T.K., Kittler, J., Cipolla, R.: Discriminative learning and recognition of image set classes using canonical correlations. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(6), 1005–1018 (2007)
7. Fan, W., Yeung, D.: Locally linear models on face appearance manifolds with application to dual-subspace based classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1384–1390 (2006)
8. Wolf, L., Shashua, A.: Learning over sets using kernel principal angles. The Journal of Machine Learning Research 4, 913–931 (2003)
9. Hamm, J., Lee, D.D.: Grassmann discriminant analysis: a unifying view on subspace-based learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 376–383. ACM (2008)
10. Wang, R., Chen, X.: Manifold discriminant analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 429–436. IEEE (2009)
11. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 711–720 (1997)
12. Bishop, C.M.: Pattern recognition and machine learning, vol. 4. Springer, New York (2006)
13. Turk, M.A., Pentland, A.P.: Face recognition using eigenfaces. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591. IEEE (1991)
14. Lee, K.C., Ho, J., Yang, M.H., Kriegman, D.: Visual Tracking and Recognition Using Probabilistic Appearance Manifolds. In: Computer Vision and Image Understanding (2005)
15. Gross, R., Shi, J.: The CMU Motion of Body (MoBo) Database. Technical Report CMU-RI-TR-01-18, Robotics Institute, Pittsburgh, PA (2001)
16. Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57(2), 137–154 (2004)
17. He, X., Niyogi, P.: Locality Preserving Projections. In: Advances in Neural Information Processing Systems (2004)

An Incremental Class Boundary Preserving Hypersphere Classifier

Noel Lopes[1,2] and Bernardete Ribeiro[1]

[1] CISUC - Department of Informatics Engineering, University of Coimbra, Portugal
[2] UDI/IPG - Research Unit, Polytechnic Institute of Guarda, Portugal
noel@ipg.pt, bribeiro@dei.uc.pt

Abstract. Recent progress in sensing, networking and data management has led to a wealth of valuable information. The challenge is to extract meaningful knowledge from such data produced at an astonishing rate. Unlike batch learning algorithms designed under the assumptions that data is static and its volume is small (and manageable), incremental algorithms can rapidly update their models to incorporate new information (on a sample-by-sample basis). In this paper we propose a new incremental instance-based learning algorithm which presents good properties in terms of multi-class support, complexity, scalability and interpretability. The Incremental Hypersphere Classifier (IHC) is tested in well-known benchmarks yielding good classification performance results. Additionally, it can be used as an instance selection method since it preserves class boundary samples.

Keywords: Incremental Learning, Classification, Machine Learning.

1 Introduction

Machine learning algorithms are commonly designed under the assumptions that data is static by nature and that its volume is small and manageable enough for them to be successfully applied in a timely manner. Usually, the emphasis is set on effectiveness (e.g. classification performance) rather than on efficiency (e.g. time required to produce a classifier) [13]. However, these two premises rarely hold true for modern databases. The continuous and rapid progress in data acquisition, networking and storage devices has led to the proliferation of data repositories that can store huge amounts of information [13,5], making most algorithms inapplicable to numerous real-world problems [8]. Rationally, when dealing with large amounts of data it is conceivable that the memory capacity will be insufficient to store every piece of relevant information during the whole learning process [4]. Moreover, even if the required memory is available, the computational requirements to process all the data in a timely manner would be prohibitive. Additionally, modern databases are dynamic by nature (i.e. they are incessantly being fed with new information). Therefore, both real-time model adaptation and classification are crucial tasks to extract valuable and

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 690–699, 2011.
© Springer-Verlag Berlin Heidelberg 2011

up-to-date information from the original data, playing a vital role in many industry segments (e.g. financial sectors, telecommunications) that rely on knowledge discovery in databases (KDD) and data mining tasks to stay competitive [9].

Reducing both the memory and the computational requirements inherent to machine learning algorithms can be accomplished by using instance selection methods that select a representative subset of the data. The idea is to identify the relevant instances for the learning process while discarding the superfluous ones. These methods can be divided in two groups (wrapper and filter) according to the strategy used for choosing the instances [6]. Unlike filter methods, wrapper methods use a selection criterion that is based on the accuracy obtained by a classifier (instances that do not contribute to improve the accuracy are discarded). A review of both wrapper and filter methods can be found in López et al. [6]. Although instance selection methods can effectively reduce the volume of data to be processed, their application may be time consuming (in particular for wrapper methods) and in some situations we may find that we are simply transferring the complexity from the learning methods to the instance selection methods. Usually, instance selection methods present scaling problems: for very large datasets the run-times may grow to the point where they become inapplicable [6].

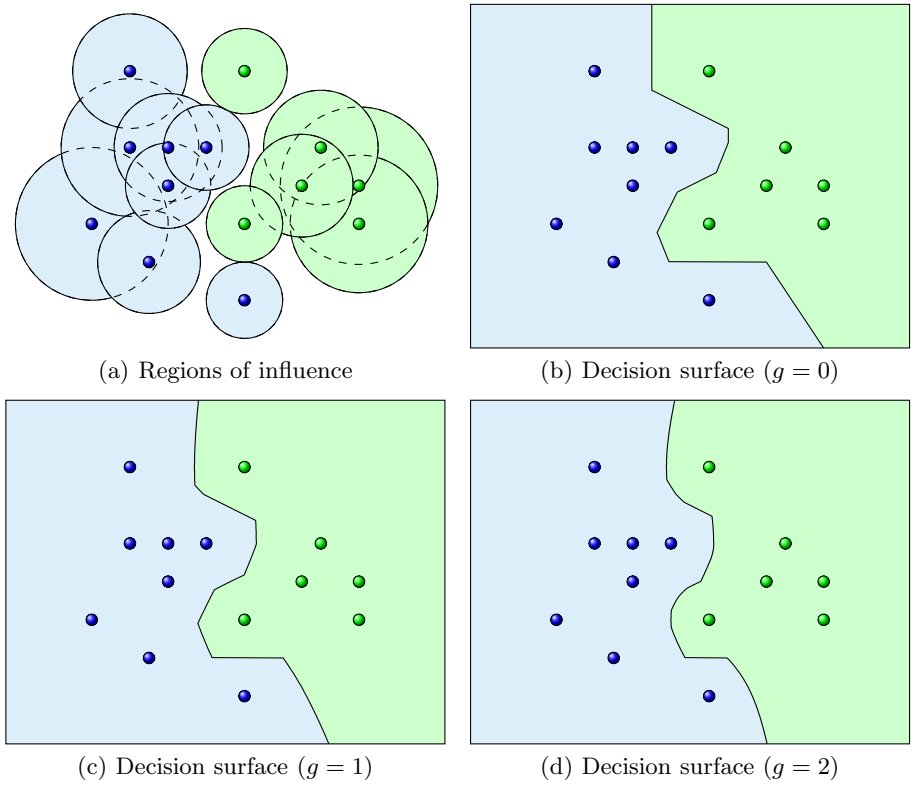
A more desirable approach to deal with the memory and computational limitations consists of using incremental learning algorithms. In this approach, the learner gradually adjusts its model as it receives new data. At each step the algorithm can only access a limited number of new samples (instances) from which a new hypothesis is built upon [4].

Another important consideration when extracting information from data repositories is the interpretability of the resulting models. In some application domains, the comprehensibility of the decisions is as valuable as having accurate models. Understanding the predictions of the models can improve the users' confidence on them [7,2].

In this paper we present a new incremental instance-based learning algorithm which exhibits advantages in terms of multi-class support, complexity, scalability and interpretability while providing good classification results. Additionally, it can be used as an instance selection method since it preserves class boundary samples. The remainder of this paper is organized as follows. The next section describes the proposed algorithm. Section 3 analyses the results. Finally, section 4 presents the conclusions and points out future lines of work.

2 Incremental Hypersphere Classifier Algorithm

The Incremental Hypersphere Classifier (IHC) algorithm is relatively simple. Its main task consists of assigning a region (zone) of influence to each sample, by which classification is achieved. Let us consider that a training sample i consists of an input vector $\mathbf{x}_i \in \mathbb{R}^d$ with an associated class label $y_i \in \{1, \ldots, C\}$, where



(a) Regions of influence (b) Decision surface ($g = 0$)

(c) Decision surface ($g = 1$) (d) Decision surface ($g = 2$)

Fig. 1. Application of the IHC algorithm to a toy problem

d is the space dimension and C is the number of classes. The region of influence of sample i is then defined by an hypersphere of radius r_i, given by (1):

$$r_i = \frac{\min(||\mathbf{x}_i - \mathbf{x}_j||)}{2}, \text{ for all } j \text{ where } y_j \neq y_i \tag{1}$$

The radius is defined so that hypersphere's belonging to different classes do not overlap. However, regions of influence belonging to the same class may overlap. Given any two samples i and j of different classes ($y_i \neq y_j$), the radiuses (r_i and r_j) will be at most half of the distance between the two points ($\mathbf{x}_i$ and $\mathbf{x}_j$), which is the maximum value that r_i and r_j can have without overlapping their hypersphere's. Figure 1(a) shows the regions of influence for a chosen toy problem.

New data points are classified according to the class of the nearest region of influence (not the nearest sample). Let $\mathbf{x}_k$ represent an input vector whose class y_k is unknown. Then, sample k belongs to class y_i ($y_k = y_i$) provided that:

$$||\mathbf{x}_i - \mathbf{x}_k|| - ga_i r_i \leq ||\mathbf{x}_j - \mathbf{x}_k|| - ga_j r_j, \text{ for all } j \neq i \tag{2}$$

where g (gravity) controls the extension of the zones of influence, increasing or shrinking them and a_i is the accuracy of sample i when classifying itself and the forgotten training samples for which i was the nearest sample in memory. A forgotten sample is a sample that either has been removed from memory or did not qualify to enter the memory. The accuracy is the first mechanism of defense against outliers. As it decreases so does the influence of the hypersphere associated. This effectively reduces the damage caused by outliers and by samples with zones of influence excessively large.

Figures 1(b), 1(c) and 1(d) show the decision surface generated by the IHC algorithm, considering different gravity values, for a toy problem. Note that for $g = 0$ the decision rule of the IHC is exactly the same as the one of the 1-NN (see eq. 2). Thus, it is interesting to point that (for $g > 0$) the former generate smoother decision surfaces (see Figures 1(b), 1(c) and 1(d)).

By analyzing the radius of the generated hypersphere's it is possible to observe that those near to the decision border have smaller radius (see Figure 1). In fact the farthest from the frontier the hypersphere is, the larger its radius will be. This provides a simple method for determining the relevance that a sample has in the classification task. When the memory is full, the radius of a new sample is compared with the radius of the nearest sample of the same class and the one with the smallest radius is kept in the memory while the other is discarded. By doing so, we are keeping the samples that play the most significant role in the construction of the decision surface (given the available memory) while removing those that have less or no impact in the model. The radius of a new sample is only compared with the one of its nearest sample to prevent the concentration of the memory samples in the same space region. Unfortunately, outliers will most likely have small radius and end-up occupying our limited memory resources. Thus, although their impact is diminished by the use of the accuracy in (2), it is still important to identify and remove them from memory. To address this problem we mimic the process used by the IB3 (Instance-Based learning) algorithm, which consists of removing all samples that are believed to be noisy by employing a significance test [12,1]. Confidence intervals are determined both for the instance accuracy (not including its own classification, unlike in eq. (2)) and for the relative frequency of the classes [1]. The instances whose maximum (interval) accuracy is less than the minimum class frequency (for the desired confidence level – typically 70%) are considered outliers and consequently dropped off.

A major advantage of this algorithm relies on the possibility of building models incrementally on a sample by sample basis. Algorithm 1 describes the main steps required to incorporate a new sample, k, on the IHC model. To cope with unbalanced datasets and avoid storing a disproportionate number of samples for each class, the algorithm assumes that the memory is divided into C groups. Considering that the available memory can hold up to N samples, the complexity of this algorithm is $O(2dN)$.

Another advantage of the algorithm is that it can accommodate the restrictions in terms of memory and computational power, creating the best model

possible for the amount of resources given, instead of requiring systems to comply with its own requirements. Since we can control the amount of memory and computational power required by the algorithm (by changing the value of N) and due to its scalability (memory and computational requirements grow linearly with the number of samples stored) it is feasible to create up-to-date models in real-time to extract meaningful information from data.

Algorithm 1. IHC algorithm

procedure IncorporateSample($\mathbf{x}_k$, y_k)
  $r_k \leftarrow \infty$ ▷ Radius of sample k
  $d \leftarrow \infty$ ▷ Distance to the nearest sample (using $||\mathbf{x}_i - \mathbf{x}_k|| - ga_i r_i$)
  $n \leftarrow$ null ▷ Nearest sample (using $||\mathbf{x}_i - \mathbf{x}_k|| - ga_i r_i$)
  $tp_k \leftarrow 1$ ▷ True positives (classified by sample k)
  $fp_k \leftarrow 1$ ▷ False positives ($a_k = \frac{tp_k}{tp_k + fp_k}$)
  for $class \leftarrow 1, \ldots, C$ **do**
    for all sample $i \subset memory[class]$ **do**
      if $||\mathbf{x}_i - \mathbf{x}_k|| - ga_i r_i < d$ **then**
        $d \leftarrow ||\mathbf{x}_i - \mathbf{x}_k|| - ga_i r_i$
        $n \leftarrow i$
      end if
      if $class \neq y_k$ **then**
        if $\frac{||\mathbf{x}_i - \mathbf{x}_k||}{2} < r_k$ **then** $r_k \leftarrow \frac{||\mathbf{x}_i - \mathbf{x}_k||}{2}$
        if $\frac{||\mathbf{x}_i - \mathbf{x}_k||}{2} < r_i$ **then** $r_i \leftarrow \frac{||\mathbf{x}_i - \mathbf{x}_k||}{2}$
      end if
    end for
  end for
  if $memory[y_k]$ **is full and** $n \neq$ null **then**
    if $r_n > r_k$ **then**
      Remove sample n from $memory[y_k]$
      $d \leftarrow \infty$
      $j \leftarrow$ null ▷ Nearest sample of sample n
      for $class \leftarrow 1, \ldots, C$ **do**
        for all sample $i \subset memory[class]$ **do**
          if $||\mathbf{x}_i - \mathbf{x}_n|| - ga_i r_i < d$ **then**
            $d \leftarrow ||\mathbf{x}_i - \mathbf{x}_n|| - ga_i r_i$
            $j \leftarrow i$
          end if
        end for
      end for
      if $j \neq$ null **then**
        if $y_n = y_j$ **then** $tp_j \leftarrow tp_j + 1$ **else** $fp_j \leftarrow fp_j + 1$
      end if
    else
      if $y_n = y_k$ **then** $tp_n \leftarrow tp_n + 1$ **else** $fp_n \leftarrow fp_n + 1$
    end if
  end if
  if $memory[y_k]$ **is not full then** Add sample k to $memory[y_k]$
end procedure

Table 1. IHC and 1-NN classification performance (F-measure (%) macro-average) for the test datasets of the UCI benchmark experiments

Database (DB)	Samples	Inputs	Classes	1-NN	IHC($g = 1$)	IHC($g = 2$)
Breast cancer (BC)	569	30	2	95.15 ± 0.41	$\mathbf{96.07 \pm 0.30}$	$\mathbf{96.45 \pm 0.36}$
Ecoli (EC)	336	7	8	66.04 ± 0.82	$\mathbf{67.51 \pm 0.72}$	$\mathbf{68.03 \pm 0.78}$
German credit data (GC)	1000	59	2	$\mathbf{64.38 \pm 0.96}$	63.98 ± 0.95	63.55 ± 0.95
Glass identification (GL)	214	9	6	68.77 ± 1.63	$\mathbf{70.30 \pm 2.20}$	$\mathbf{69.81 \pm 2.23}$
Haberman's survival (HA)	306	3	2	55.53 ± 2.04	55.26 ± 2.35	$\mathbf{56.36 \pm 1.92}$
Heart - Statlog (HE)	270	20	2	75.30 ± 1.60	$\mathbf{75.92 \pm 1.28}$	$\mathbf{76.19 \pm 1.27}$
Ionosphere (IO)	351	34	2	85.90 ± 0.69	$\mathbf{90.98 \pm 0.54}$	$\mathbf{92.55 \pm 0.47}$
Iris (IR)	150	4	3	95.70 ± 0.69	$\mathbf{95.71 \pm 0.61}$	$\mathbf{96.04 \pm 0.64}$
Pima diabetes (PD)	768	8	2	66.95 ± 1.06	$\mathbf{68.41 \pm 1.00}$	$\mathbf{70.09 \pm 0.97}$
Sonar (SO)	208	60	2	85.60 ± 1.76	$\mathbf{85.63 \pm 1.79}$	$\mathbf{87.03 \pm 1.50}$
Tic-Tac-Toe (TT)	958	9	2	49.47 ± 0.47	$\mathbf{73.43 \pm 0.54}$	$\mathbf{81.21 \pm 0.83}$
Vehicle (VE)	946	18	4	69.35 ± 0.76	$\mathbf{69.46 \pm 0.71}$	68.78 ± 0.93
Wine (WI)	178	13	3	95.90 ± 0.51	$\mathbf{96.80 \pm 0.44}$	$\mathbf{96.93 \pm 0.64}$
Yeast (YE)	1484	8	10	56.32 ± 1.04	$\mathbf{57.73 \pm 1.12}$	$\mathbf{58.75 \pm 0.86}$

3 Experimental Results

To evaluate the performance of the IHC, we carried out extensive experiments on several UCI databases [3] with different characteristics (number of samples, features and classes). For statistical significance each experiment was run 30 times using different random 5-fold stratified cross validation partitions. Since the IHC is an instance based classifier, we choose to compare it with the well-known 1-NN that has demonstrated good classification performance in a wide range of real-world problems [10]. Table 1 reports the F-measure performance for both the baseline 1-NN and for the IHC using parameters $g = 1$ and $g = 2$ (no memory restrictions were imposed).The IHC algorithm excels the 1-NN in all the experiments except in the German credit data (where the 1-NN presents slightly better results). Moreover, in the case of the tic-tac-toe the IHC performs considerably better (with an F-Measure discrepancy of 31.74% for $g = 2$). To validate the referred improvements, we conducted the Wilcoxon signed rank test. The null hypothesis of the 1-NN having an equal or better F-Measure than the IHC algorithm is rejected at a significance level of 0.005 (both for $g = 1$ and $g = 2$). Thus, there is strong evidence that the IHC significantly outperforms the 1-NN. Given the good results obtained, a particular area in which the IHC algorithm may be useful is on the development of ensembles of classifiers. The ensemble is a system that incorporates different individual models (created by using one or several machine learning algorithms), combining their outputs to produce a single answer for a specific problem [11]. The rationale behind this approach is to improve the quality of the solution produced by combining different imperfect models to produce a system that can excel the individual ones. Ensembles can present higher classification rates than a single best classifier, provided that the integrated classifiers are diverse and accurate [10].

While the results obtained so far demonstrate the usefulness of the IHC algorithm (see Table 1), we are particularly interested in its behavior against limited

Table 2. Classification performance (F-measure (%) macro-average) and compression ratio (%) of the IHC and IB3 algorithms for the UCI benchmark experiments

	Compression ratio		F-Measure (test)		F-Measure (overall)	
DB	**IB3**	**IHC**	**IB3**	**IHC**	**IB3**	**IHC**
BC	93.80 ± 1.18	$\mathbf{93.89 \pm 0.07}$	93.47 ± 1.02	$\mathbf{93.64 \pm 0.97}$	94.35 ± 0.66	$\mathbf{94.66 \pm 0.70}$
EC	$\mathbf{78.98 \pm 2.23}$	78.73 ± 0.22	63.80 ± 3.41	$\mathbf{65.30 \pm 1.88}$	60.96 ± 3.28	$\mathbf{83.06 \pm 1.58}$
GC	95.30 ± 1.29	$\mathbf{95.38 \pm 0.12}$	55.91 ± 2.20	$\mathbf{56.33 \pm 2.05}$	60.17 ± 1.78	$\mathbf{61.49 \pm 0.84}$
GL	$\mathbf{86.57 \pm 2.40}$	86.04 ± 0.22	35.63 ± 2.19	$\mathbf{51.43 \pm 3.04}$	41.50 ± 3.19	$\mathbf{62.05 \pm 1.68}$
HA	$\mathbf{97.45 \pm 1.13}$	97.12 ± 0.30	44.85 ± 2.96	$\mathbf{54.10 \pm 3.84}$	46.21 ± 3.62	$\mathbf{57.33 \pm 2.14}$
HE	92.44 ± 1.64	$\mathbf{92.76 \pm 0.24}$	$\mathbf{79.53 \pm 1.58}$	76.68 ± 2.39	$\mathbf{80.72 \pm 0.82}$	79.60 ± 1.62
IO	93.41 ± 2.37	$\mathbf{93.64 \pm 0.08}$	75.21 ± 4.48	$\mathbf{81.04 \pm 2.77}$	77.30 ± 3.81	$\mathbf{82.87 \pm 2.59}$
IR	81.44 ± 3.28	$\mathbf{82.50 \pm 0.00}$	$\mathbf{93.87 \pm 1.68}$	93.60 ± 1.78	94.55 ± 1.10	$\mathbf{95.92 \pm 1.20}$
PD	$\mathbf{94.04 \pm 1.41}$	94.03 ± 0.15	$\mathbf{66.60 \pm 2.33}$	64.68 ± 1.81	$\mathbf{69.22 \pm 1.78}$	68.01 ± 1.10
SO	$\mathbf{97.63 \pm 1.44}$	97.62 ± 0.07	48.26 ± 7.37	$\mathbf{60.62 \pm 4.83}$	49.71 ± 7.26	$\mathbf{62.05 \pm 3.39}$
TT	$\mathbf{93.99 \pm 2.25}$	93.88 ± 0.11	61.56 ± 3.80	$\mathbf{61.99 \pm 1.57}$	63.81 ± 4.00	$\mathbf{66.67 \pm 0.85}$
VE	$\mathbf{85.48 \pm 1.39}$	85.23 ± 0.05	$\mathbf{62.26 \pm 1.13}$	60.90 ± 1.74	68.15 ± 0.94	$\mathbf{68.20 \pm 1.05}$
WI	82.48 ± 1.99	$\mathbf{83.15 \pm 0.16}$	$\mathbf{94.03 \pm 1.28}$	93.22 ± 1.43	94.93 ± 0.88	$\mathbf{95.59 \pm 0.84}$
YE	82.68 ± 1.38	$\mathbf{82.90 \pm 0.09}$	37.52 ± 3.68	$\mathbf{47.21 \pm 1.25}$	45.00 ± 4.51	$\mathbf{61.70 \pm 0.90}$

memory and processing power resources. In such scenarios it is up to the algorithm to decide what is relevant and what is accessory (or less relevant). Clearly, there is a trade-off between the amount of information stored and the performance of the resulting models. To exacerbate this problem, the order in which samples are presented may exert a profound impact on the algorithms decisions. Different orders impose distinct bias, affecting the algorithm results.

Table 2 compares the performance of the IHC algorithm (for $g = 1$) with the IB3 algorithm. The latter is one of the most successful instance selection and instance-based learning standard algorithms [8], presenting low storage requirements and high accuracy results [12]. Moreover, IB3 is an incremental algorithm, making it the ideal candidate for comparison purposes. For fairness and unbiased comparison, the order in which samples were presented was the same for both algorithms. It is not possible to anticipate the amount of memory that IB3 will require for a given problem. In this aspect the IHC algorithm is advantageous since it respects the memory bounds imposed. Hence, we configured the memory requirements to match (as closely as possible) those of the IB3 algorithm. On average IB3 presents a compression ratio of 89.69% while the IHC presents a compression ratio of 89.78%. In terms of performance on the test datasets the IHC excels the IB3 in 9 of the 14 benchmarks. On average the IHC algorithm improves the F-Measure by 3.45% relatively to the IB3. The performance gap is specially appreciable in the glass, Haberman's, sonar and yeast benchmarks where the F-Measure is improved respectively by 15.80%, 9.25%, 12.36% and 9.69%.

Real world databases often present a high-degree of redundancy. In some situations similar (or identical) records may be frequent. Therefore it is important to determine the performance of the algorithms on the forgotten data. To this end, Table 2 includes the overall performance (train and test data). With respect to the aforementioned, we reject the null hypothesis of IB3 having an equal or better

expected F-measure at a significance level of 0.005 for the Wilcoxon signed rank test. Hence, there is strong statistical evidence that the IHC algorithm preserves more (better) information of the forgotten samples than the IB3 algorithm, thus yielding superior results. This is accomplished by using the information of each sample that was ever presented to update the model (i.e. the radius of the samples of different classes in the memory). On average the IHC algorithm improves the F-Measure by 6.62%. Moreover, in the ecoli, glass and yeast benchmarks the performance gap is particularly evident with a respective improvement of 22.10%, 20.55%, 16.70%. It is worth to mention that the IHC results could be enhanced by fine-tuning the value of g.

To analyze the behavior of the IHC algorithm in a large database, we apply it to the KDD Cup 1999 dataset (available at the UCI repository) which contains approximately 5 million of samples. The objective consists of building a network intrusion detector capable of distinguishing between normal connections and 4 different types of attacks (denial of service (DOS), unauthorized access from a remote machine (R2L), unauthorized access to local superuser privileges (U2R) and probing). To discriminate between the 5 classes, our model uses 40 features. In real-world scenarios we cannot control the order in which samples appear, thus they were presented to the algorithm in the same order as they appear on the dataset. Figures 2 and 3 show respectively the time required to update the model and the accuracy according to the memory used by the algorithm. As expected the time necessary to update the model grows linearly with the amount of memory used, making the IHC a highly scalable algorithm. To update the model requires approximately 4 milliseconds for $N = 50000$, using a Core 2 Quad Q 9300 2.5 GHz CPU. This demonstrates real-time model adaptability and knowledge extraction are feasible. The accuracy depends substantially on the amount of memory supplied to the algorithm. Lower memory bounds imply larger oscillations on the accuracy (see Figure 3). These occur when samples conveying information that is not yet covered by the model concepts are presented to the algorithm. In this problem, the first 7448 samples belong to the normal class and within the first 75985 only 4 samples belong to another class (U2R). At this point the model concepts do not account for any other classes and samples from other classes will be misclassified. As a result a sudden decrease in the models accuracy is experienced. Eventually, when the new concepts are integrated in the model, the accuracy recovers. Rationally, if the memory footprint is too small we may find ourselves in a position where there is not enough information to separate useful from accessory information. For example, for $N = 50$ only 10 samples per class can be stored. Given that the first 7448 samples belong all to the same class, there is no way for the algorithm to make the correct (informed) decision of which samples to retain in the memory. Of course the larger the number of samples the algorithm is allowed to store the greater the chances it has to preserve those lying near the decision border. Therefore, lower memory bounds result in accentuated oscillations and in a reduced overall accuracy (as depicted in Figure 3). Nevertheless, the algorithm presents a good classification performance even with as little memory as the necessary to store 50 samples.

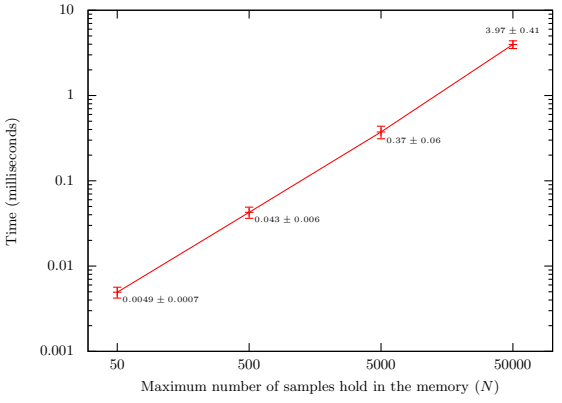


Fig. 2. Average time required to update the IHC model (after presenting a new sample) for the KDD Cup 1999 dataset

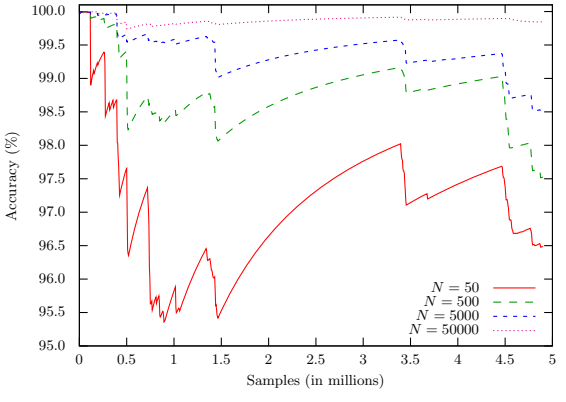


Fig. 3. Accuracy of the IHC model for the KDD Cup 1999 dataset

4 Conclusions and Future Work

Extracting real-time information from large and dynamic data repositories is a very hard task, which renders most machine learning algorithms inapplicable. Nevertheless, the potential benefits and the competitive advantages that can be obtained justify the design of algorithms for that specific purpose. In this context, we presented a new incremental and highly-scalable algorithm with multi-class support that can accommodate memory and computational restrictions, creating the best model possible for the amount of resources given. The experiments demonstrate its ability to update the models and classify new data in real-time, while maintaining superior classification performance. Additionally, the resulting models are interpretable making this algorithm useful even in domains where interpretability is a key factor. Finally, since it keeps the instances

(that are believed to be lying) on the decision frontier, it can also be an optimal choice for selecting a representative subset of the data for more sophisticated algorithms.Due to its capacity to minimize the impact of noisy samples, eventually removing them from memory, the algorithm can handle concept-drift scenarios. Future work will compare this ability with other algorithms designed specifically for that purpose. Another line of work consists of evaluating the impact of using different values of g for distinct classes as well as adjusting them automatically.

Acknowledgment. FCT (Foundation for Science and Technology) is gratefully acknowledged for funding the project FCOMP-01-0124-FEDER-010160.

References

1. Aha, D., Kibler, D., Albert, M.: Instance-based learning algorithms. Machine Learning 6(1), 37–66 (1991)
2. Bibi, S., Stamelos, I.: Selecting the appropriate machine learning techniques for the prediction of software development costs. In: Proc. of Artificial Intelligence Applications & Innovations, pp. 533–540 (2006)
3. Frank, A., Asuncion, A.: UCI machine learning repository, http://archive.ics.uci.edu/ml
4. Jain, S., Lange, S., Zilles, S.: Towards a better understanding of incremental learning. In: Balcázar, J.L., Long, P.M., Stephan, F. (eds.) ALT 2006. LNCS (LNAI), vol. 4264, pp. 169–183. Springer, Heidelberg (2006)
5. Liu, H., Motoda, H.: On issues of instance selection. Data Mining and Knowledge Discovery 6, 115–130 (2002)
6. López, J., Ochoa, J., Trinidad, J., Kittler, J.: A review of instance selection methods. Artificial Intelligence 34, 133–143 (2010)
7. Pappa, G., Freitas, A.: Automating the Design of Data Mining Algorithms: An Evolutionary Computation Approach. Springer, Heidelberg (2010)
8. Pedrajas, N., Castillo, J., Boyer, D.: A cooperative coevolutionary algorithm for instance selection for instance-based learning. Machine Learning 78, 381–420 (2010)
9. Reinartz, T.: A unifying view on instance selection. Data Mining and Knowledge Discovery 6(2), 191–210 (2002)
10. Tahir, A., Smith, J.: Creating diverse nearest-neighbour ensembles using simultaneous metaheuristic feature selection. Pattern Recognition Letters 31, 1470–1480 (2010)
11. Wang, W.: Some fundamental issues in ensemble methods. In: Proc. of the International Joint Conference on Neural Networks, pp. 2243–2250 (2008)
12. Wilson, D., Martinez, T.: Reduction techniques for instance-based learning algorithms. Machine Learning 38(3), 257–286 (2000)
13. Zhou, Z.: Three perspectives of data mining. Artificial Intelligence 143(1), 139–146 (2003)

Co-clustering for Binary Data with Maximum Modularity

Lazhar Labiod and Mohamed Nadif

LIPADE, Université Paris Descartes,
45, rue des Saints Pères, 75006 Paris, France
{firstname.lastname}@parisdescartes.fr

Abstract. The modularity measure have been recently proposed for graph clustering which allows automatic selection of the number of clusters. Empirically, higher values of the modularity measure have been shown to correlate well with graph clustering. In order to tackle the co-clustering problem for binary data, we propose a generalized modularity measure and a spectral approximation of the modularity matrix. A spectral algorithm maximizing the modularity measure is then presented to search for the row and column clusters simultaneously. Experimental results are performed on a variety of real-world data sets confirming the interest of the use of the modularity.

Keywords: modularity, binary data, co-clustering.

1 Introduction

Cluster analysis, or data clustering is an important tool in a variety of scientific areas including pattern recognition, information retrieval, micro-arrays and data mining, is a family of exploratory data analysis methods which can be used to discover structures in data. The aim of clustering is to organize the set of objects into homogeneous clusters. To define the notion of homogeneity, we often use similarity or dissimilarity measures on this set. It is a property designed to ensure greater similarity between two objects in the same cluster than between two objects in different clusters. Note that in practice, this objective is impractical but by optimizing a partitional clustering objective we can define a partition of a set of objects into clusters, such that the objects in a cluster are more similar to each other than to objects in other clusters.

Although many clustering procedures such as hierarchical clustering and k-means aim to construct an optimal partition of objects or, sometimes, variables, there are other methods, known as block clustering methods, which consider the two sets simultaneously and organize the data into homogeneous blocks. In recent years block clustering, also denoted co-clustering or bi-clustering, has become an important challenge in the context of data mining. In the text mining field, Dhillon [10] has proposed a spectral block clustering method by exploiting the duality between rows (documents) and columns (words). In the analysis of microarray data, where data are often presented as matrices of expression levels

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 700–708, 2011.
© Springer-Verlag Berlin Heidelberg 2011

of genes under different conditions, block clustering of genes and conditions has been used to overcome the problem of choosing the similarity on the two sets found in conventional clustering methods [3]. The aim of block clustering is to try to summarize this matrix by homogeneous blocks. A wide variety of procedures have been proposed for finding patterns in data matrices. These procedures differ in the pattern they seek, the types of data they apply to, and the assumptions on which they rest. In particular we should mention the work of [9], [8], [12] who have proposed some algorithms dedicated to different kinds of matrices. The basic idea of these methods consist in making permutations of objects and attributes in order to draw a correspondence structure between these two sets. Hereafter, we illustrate the co-clustering aim by an example which consists of the characteristics (rows) and 16 townships (columns), each cell indicates the presence 1 or absence 0 of a characteristic on a township (Table 1). This example has been used by Niermann in [15] for data ordering task and the author aims to reveal a block diagonal form (Table 2). Obviously, Table 2 is preferable with respect to conciseness. Then, we can characterize each cluster of townships by a cluster of characteristics that we report in Table 3. To achieve this goal we consider a new approach based on the modularity measure. This

Table 1. The nine characteristics of 16 townships {A,. . . ,P}

Characteristics	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
High School	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
Agricult Coop	0	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0
Rail station	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
One Room School	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	1
Veterinary	0	1	1	1	0	0	1	0	0	0	0	1	0	0	1	0
No Doctor	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	1
No Water Supply	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0
Police Station	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
Land Reallocation	0	1	1	1	0	0	1	0	0	0	0	1	0	0	1	0

Table 2. Reorganization of the townships and characteristics after co-clustering

Characteristics	H	K	B	C	D	G	L	O	M	N	J	I	A	P	F	E
High School	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Railway Station	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Police Station	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Agricult Coop	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Veterinary	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Land Reallocation	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
One Room School	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
No Doctor	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
No Water Supply	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

Table 3. Characterization of each cluster of townships by a cluster of characteristic

Class of townships	Class of characteristics
{H, K}	{High School, Railway Station,Police Station}
{B, C, D, G, L, O}	{Agricult Coop, Veterinary,Land Reallocation}
{M, N, J, I, A, P, F, E}	{ One Room School, No Doctor, No Water Supply}

measure has been used recently for graph clustering [6] [1]. In this paper we show how Newman's modularity measure can be generalized to binary data co-clustering and can be related to the broader family of spectral clustering methods. Specifically:

- We propose a new generalized modularity measure for binary data coclustering.
- We show how the problem of maximizing the generalized modularity measure $\tilde{Q}$ can be reformulated as an eigenvector problem. In this manner we link work on binary data co-clustering using the generalized modularity measure to relevant work on spectral co-clustering [10].
- We develop an efficient spectral based procedure to find the optimal simultaneous objects and attributes partitions maximizing the normalized modularity criterion.
- Unlike to known spectral clustering methods, we show that the modularity measure allows a natural co-clusters identification, i.e. the maximum value of modularity correlate well with the optimal number of co-clusters.

The rest of the paper is organized as follows: Section 2 provides the proposed generalized modularity measure. Some discussions on the spectral connection and optimization procedure are given in Section 3. Section 4 is devoted to numerical experimental results. Finally, Section 5 presents the conclusion and some future works.

Notation. We will consider in the rest of this paper, the partition of the sets I of objects and the set J of attributes into g non overlapping clusters, where g may be greater or equal to 2. Let us define an $N \times g$ index matrix R and an $M \times g$ index matrix C with one column for each cluster; $R = (R_1|R_2|\ldots|R_g)$ and $C = (C_1|C_2|\ldots|C_g)$. Each column R_k or C_k is defined as follows: $r_{ik} = 1$ if object i belongs to cluster R_k and $r_{ik} = 0$ otherwise, and in the same manner $c_{jk} = 1$ if attribute j belongs to cluster C_k and $c_{jk} = 0$ otherwise.

2 Generalized Modularity Measure

This section shows how to adapt the Modularity measure for binary data co-clustering. But before, we firstly review the modularity in graph clustering task.

2.1 Modularity and Graphs

Modularity is a recent quality measure for graph clustering, it has immediately received a considerable attention in several disciplines [6] [1]. Maximizing the modularity measure can be expressed in the form of an integer linear programming. Given the graph $G = (V, E)$, let A be a binary, symmetric matrix with (i, i') as entry; and $a_{ii'} = 1$ if there is an edge between the nodes i and i'. If there is no edge between nodes i and i', $a_{ii'}$ is equal to zero. We note that in our problem, A is an input having all information on the given graph G and

is often called an adjacency matrix. Finding a partition of the set of nodes V into homogeneous subsets leads to the resolution of the following integer linear program: $\max_X Q(A,X)$ where $Q(A,X)$ is the modularity measure

$$Q(A,X) = \frac{1}{2|E|}\sum_{i,i'=1}^{n}(a_{ii'} - \frac{a_{i.}a_{i'.}}{2|E|})\sum_{k=1}^{g} r_{ik}r_{i'k}.$$

Taking $x_{ii'} = \sum_{k=1}^{g} r_{ik}r_{i'k}$, the expression of Q becomes

$$Q(A,X) = \frac{1}{2|E|}\sum_{i,i'=1}^{n}(a_{ii'} - \frac{a_{i.}a_{i'.}}{2|E|})x_{ii'}, \tag{1}$$

where $2|E| = \sum_{i,i'} a_{ii'} = a_{..}$ is the total number of edges and $a_{i.} = \sum_{i'} a_{ii'}$ the degree of i. Let $\delta = (\delta_{ii'})$ be the $(n \times n)$ data matrix defined by $\forall i, i'\ \ \delta_{ii'} = \frac{a_{i.}a_{i'.}}{2|E|}$, the expression (1) becomes $Q(A,X) = \frac{1}{2|E|}Tr[(A-\delta)X]$. The researched binary matrix X is defined by RR^t which models a partition in a relational space and therefore must check the properties of an equivalence relation.

2.2 Modularity Measure for Binary Data

The basic idea consists in modelling the simultaneous row and column partitions using a block seriation relation Z defined on $I \times J$. Noting that $Z = RC^t$ then the general term can be expressed as $z_{ij} = \sum_{k=1}^{g} r_{ik}c_{jk}$; $z_{ij} = 1$ if (i,j) belongs to the block k and $z_{ij} = 0$ otherwise. Now, given a rectangular matrix A defined on $I \times J$, to tackle the co-clustering for binary data, we propose the following generalized modularity measure $Q_1(A,Z) = \frac{1}{2|E|}\sum_{i,j=1}^{n}(a_{ij} - \frac{a_{i.}a_{.j}}{2|E|})z_{ij}$. where $2|E| = \sum_{i,j} a_{ij} = a_{..}$ is the total weight of edges and $a_{i.} = \sum_j a_{ij}$ - the degree of i and $a_{.j} = \sum_i a_{ij}$- the degree of j. This Modularity measure takes the following form

$$Q_1(A,Z) = \frac{1}{2|E|}Tr[(A-\delta)^t Z]. \tag{2}$$

The matrix Z represents a block seriation relation then it must respect certain properties: binarity, assignment constraints and triad impossible (see for instance, [14]). As the objective function (2) is linear with respect to Z and as the constraints that Z must respect are linear equations, theoretically we can solve the problem using an integer linear programming solver. However, this problem is NP hard, we then use heuristics, in practice, for dealing with large data set.

3 Maximization of Normalized Generalized Modularity

The expression (2) is not balanced by the row and column cluster size, meaning that a cluster might become small when affected by outliers. Thus we propose a

new measure which we call normalized generalized modularity whose objective function is given as follows:

$$\tilde{\mathcal{Q}}_1(A,Z) = Tr[(K-\delta)^t R G^{\frac{-1}{2}} F^{\frac{-1}{2}} C^t] = Tr[(K-\delta)^t \tilde{Z}]. \quad (3)$$

where $G = diag(R^t\mathbb{1})$ is a g by g diagonal matrix, each diagonal element g_{kk} corresponds to the number of objects in the row cluster k and $F = diag(C^t\mathbb{1})$, each diagonal element f_{kk} gives the number of attributes in the column cluster k. Finally, $\mathbb{1}$ is the vector of the appropriate dimension which all its values are 1. On the other hand, note that the expression (3) can be written as $\tilde{\mathcal{Q}}_1(A,Z) = Tr[(A-\delta)^t\tilde{Z}]$. where $\tilde{Z} = \tilde{R}\tilde{C}^t$ with $\tilde{R} = RG^{\frac{-1}{2}}$ and $\tilde{C} = CF^{\frac{-1}{2}}$. It is easy to verify that $\tilde{R}$ and $\tilde{C}$ satisfy the orthogonality constraint i.e. $\tilde{R}^t\tilde{R} = I_g$ and $\tilde{C}^t\tilde{C} = I_g$, then the maximization of the normalized generalized modularity is equivalent to the following trace optimization problem $\max_{\tilde{R}^t\tilde{R}=I_g,\tilde{C}^t\tilde{C}=I_g} Tr[\tilde{R}(A-\delta)\tilde{C}^t]$. This optimization problem can be performed by Lagrange multipliers into eigenvalue problem.

3.1 Spectral Connection

In the following, the number of clusters g on I and J is assumed fixed. We use the following strategy to address the problem of finding a simultaneous partitioning that maximizes $\tilde{Q}_1(A,Z)$ as follows:1) Approximate the resulting assignment problem by relaxing it to a continuous one which can be solved analytically using eigen-decomposition techniques. 2) Compute the first $(g-1)$ left and right eigenvectors of this solution to form a $(g-1)$-dimensional embedding of data into a Euclidean space. Then we use a hard-assignment thanks to $k$means on this new space to obtain a simultaneous clustering R and C.

Proposition. Let K be a disjunctive matrix, taking $D_r = diag(A\mathbb{1})$ and $D_c = diag(A^t\mathbb{1})$, the modularity matrix $(A-\delta)$ can be approximated by the $(g-1)$th largest eigenvectors of the scaled matrix $\tilde{A} = D_r^{\frac{-1}{2}} A D_c^{\frac{-1}{2}}$ minus the trivial vectors (corresponding to the largest eigenvalue).

Proof. Note that we can rewrite A as $A = D_r^{\frac{1}{2}}(D_r^{\frac{-1}{2}} A D_c^{\frac{1}{2}})D_c^{\frac{1}{2}}$. It is well known that the largest eigenvalue of $\tilde{A} = D_r^{\frac{-1}{2}} A D_c^{\frac{-1}{2}}$ is equal to $\lambda_0 = 1$ and the associated left and right eigenvectors are respectively $U_0 = \frac{D_r^{\frac{1}{2}}\mathbb{1}}{\sqrt{a_{..}}}$ and $V_0 = \frac{D_c^{\frac{1}{2}}\mathbb{1}}{\sqrt{a_{..}}}$ [7]. Applying the spectral decomposition of the scaled matrix $\tilde{A}$ instead on A directly, leading to

$$A = D_r^{\frac{1}{2}} \sum_{k\geq 0} U_k \lambda_k V_k^t D_c^{\frac{1}{2}}. \quad (4)$$

Subtract the trivial eigenvectors corresponding to the largest eigenvalue $\lambda_0 = 1$ give $A = \frac{D_r\mathbb{1}\mathbb{1}^t D_c}{a_{..}} + D_r^{\frac{1}{2}} \sum_{k\geq 1} U_k\lambda_k V_k^t D_c^{\frac{1}{2}}$. Keeping the $(g-1)$th first eigenvectors, we obtain the following approximation $A - \frac{D_r\mathbb{1}\mathbb{1}^t D_c}{a_{..}} \approx \sum_{k=1}^{g-1} \tilde{U}_k\lambda_k\tilde{V}_k^t$ where $\tilde{U}_k = D_r^{\frac{-1}{2}} U_k$ and $\tilde{V}_k = D_c^{\frac{-1}{2}} V_k$. Then taking $\delta = \frac{D_r\mathbb{1}\mathbb{1}^t D_c}{a_{..}}$, we can approximate

$(A-\delta)$ by $\sum_{k=1}^{g-1} \tilde{U}_k \lambda_k \tilde{V}_k^t$. Furthermore, note that the general term of δ is defined by $\delta_{ij} = \frac{a_{i.} a_{.j}}{a_{..}}$, that is its expression in (2.2).

The modularity matrix $(A-\delta)$ used in (3) is expressed in terms of $(g-1)$th first eigenvectors of the scaled matrix $\tilde{A}$. Then we have a $(N \times (g-1))$ matrix $U = [U_1, ..., U_{g-1}]$ formed by the $(g-1)$ left eigenvectors and a $(M \times (g-1))$ matrix $V = [V_1, ..., V_{g-1}]$ formed by the $(g-1)$ right eigenvectors. We then normalize U into $\tilde{U}$ in which $\tilde{U}_k = \frac{D_r^{\frac{1}{2}} U_k}{||D_r^{\frac{1}{2}} U_k||}$, and V into $\tilde{V}$ in which $\tilde{V}_k = \frac{D_c^{\frac{1}{2}} V_k}{||D_c^{\frac{1}{2}} V_k||}$.

3.2 Spectral Co-clustering Algorithm

The eigenmatrices $\tilde{U}$ and $\tilde{V}$ can be an input of the $k$means or other clustering algorithms via the following new matrix $Q = \begin{pmatrix} \tilde{U} \\ \tilde{V} \end{pmatrix}$ of size $((N+M) \times g)$; a superposition of the two matrices $\tilde{U}$ and $\tilde{V}$. The term "individuals (rows)-variables (columns)" is to be taken here with a very broad sense. Indeed, the principle of superposition plays on the lack of distinction between these notions when taking into account data, to be restored at the level of the final solution. The deducted set of objects to cluster is $L = I \cup J$, union of the two sets of departure. In the matrix Q, individuals and variables are now playing a similar role, so we will refer to by the single term "object". We thus find the problem of one side clustering, since the problem is again to cluster a set of objects, this time however, the set in question is no longer either I or J, but the union of the two sets. The solution is a partition of L, we denote by $\begin{pmatrix} R \\ C \end{pmatrix}$ the corresponding matrix partition. It seeks to bring together, in homogeneous clusters the most similar objects (rows and/or columns). The proposed algorithm called *SpecCo* begins by computing the first $(g-1)$ eigenvectors ignoring the trivial ones. This algorithm is similar in spirit to the one developed by Dhillon [10]. The algorithm embed the input data into the Euclidean space by eigen-decomposing a suitable affinity matrix and then cluster Q using a geometric clustering algorithm. Hereafter, the pseudo code of the proposed algorithm.

The *SpecCo* algorithm contains two majors components: computing the eigenvectors and executing $k$means to partition the rows and columns data. We run $k$means on Q; each row is a $(g-1)$ vector. Standard $k$means with Euclidean distance metric has time complexity $O((N+M)dkt)$, where $(N+M)$ is the number of data points plus the number of attributes, and t is the number of iterations required for $k$means to converge. In addition, for the *SpecCo* algorithm there is the additional complexity for computing the matrix eigenvectors Q. For computing the largest eigenvectors using the power method or Lanczos method [13], the running time is $O(N^2 M)$ per iteration. Similar to other spectral graph clustering method, the time complexity of *SpecCo* can be significantly reduced if the affinity matrix A is sparse.

Algorithm 1. SpecCo

Input: data A, number of clusters g
Output: partition matrices R and C
1. Form the affinity matrix A
2. Define D_r and D_c to be $D_r = diag(A\mathbb{1})$ and $D_c = diag(A^t\mathbb{1})$
3. Find U,V the $(g-1)$ left-right largest eigenvectors of $\tilde{A} = D_r^{-\frac{1}{2}} A D_c^{-\frac{1}{2}}$
4. From U and V, form the matrices $\tilde{U}$, $\tilde{V}$ and $Q = \begin{pmatrix} \tilde{U} \\ \tilde{V} \end{pmatrix}$
5. Cluster the rows of Q into g clusters by using $k$means
6. Assign object i to cluster R_k if and only if the corresponding row Q_i of the matrix Q was assigned to cluster R_k and assign attribute j to cluster C_k if and only if the corresponding row Q_j of the matrix Q was assigned to cluster C_k.

4 Numerical Experiments

A performance study has been conducted to evaluate the behavior of our method *SepcCo*. We observed that most co-clustering algorithms require the number of co-clusters as an input parameter. First, we evaluate the ability of the modularity to indicate the good number of hidden co-clusters in binary data. In our experiments, we co-cluster the 16 townships data set into different number of co-clusters varying from 2 to 9. For each fixed number of co-clusters, the co-clustering modularity is computed and the optimal number of co-clusters is considered to correlate well with the maximum modularity value. Second, to test the clustering performance of our algorithm against other algorithms, the competitive retained algorithms are k modes [5], NMF and ONMTF developed in [8]. To demonstrate the ability of the modularity measure to detect the suitable number of co-clusters, we ran our algorithm on the 16 townships characteristics presented in table 1, we remark that the maximum modularity is achieved with a number of co-clusters equal to 3.

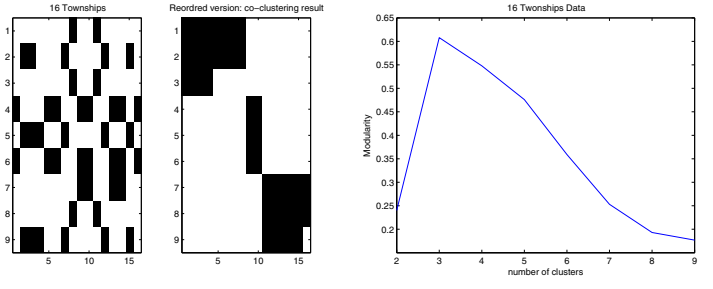


Fig. 1. left: 16 Townships data set-Middle: Reordered version - Right: Modularity versus the number of co-clusters

4.1 Evaluation of *SpecCo*

A performance study has been conducted to evaluate our method *SepcCo*. To test its clustering performance against other algorithms, we ran our algorithm

on real-life data sets. The competitive retained algorithms are *k*modes [5], NMF and ONMTF developed in [8]. The update rules of NMF are defined with the row and column coefficients matrices $\mathbf{U}$ and $\mathbf{V}$ corresponding to the following approximation $A \approx \mathbf{U}\mathbf{V}^t$, and for ONMTF with the row, column coefficients matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{S}$ ($\mathbf{S}$ consists to absorb the scales of U, V and A) corresponding to the following approximation $A \approx \mathbf{U}S\mathbf{V}^t$. These update rules of the three factors are available in [8].

Validating clustering results is a non-trivial task. In the presence of true labels, the clustering accuracy is employed to measure the quality of clustering. We focus on the quality of row clusters. Clustering Accuracy noted *Acc* discovers the one-to-one relationship between obtained clusters and true classes. It measures the extent to which each cluster contained data points from the corresponding class; it is defined as follows: $Acc = \frac{1}{N} max[\sum_{\mathcal{C}_k,\mathcal{L}_m} T(\mathcal{C}_k, \mathcal{L}_m)]$, where $\mathcal{C}_k$ is the kth cluster in the final results, and $\mathcal{L}_m$ is the true mth class. $T(\mathcal{C}_k, \mathcal{L}_m)$ is the number of entities which belong to class m and are assigned to cluster k. Accuracy computes the maximum sum of $T(\mathcal{C}_k, \mathcal{L}_m)$ for all pairs of clusters and classes, and these pairs have no overlaps. The greater clustering accuracy means, the better clustering performance.

4.2 Real Data Sets

We used five datasets for document clustering. Classic30, Classic150 are an extract of Classic3 [10] which contains three classes denoted Medline, Cisi, Cranfield as their original database source. Classic30 consists of 30 random documents described by 1000 words and Classic150 consists of 150 random documents described by 3625 words. Finally, NG2 (2 classes of documents), NG5 (5 classes) and NG10 (10 classes) are a subset of 20-Newsgroup data NG20 and composed by 500 documents described by 2000 words. These co-occurrence tables are converted to binary data; each cell higher to 0 is considered as equal to 1.

Table 4. Clustering accuracy by *k*modes, NMF, ONMTF and *SpecCo*

Datasets	kmodes	NMF	ONMTF	*SpecCo*
C30	70	73.33	70	**96.66**
C150	49.5	48.6	75.3	**86**
NG2	60	62.8	62.6	**77.4**
NG5	48.5	49.89	46.7	**58.71**
NG10	48	47	47.90	**56.71**

The obtained accuracy by the four methods are reported in table 4 showing that *SpecCo* outperforms the others methods. This good performance can be explained by the fact that the used datasets are very sparse, for which the modularity measure is a well adapted clustering criterion. Also, the algorithms have been applied to the data without prior normalization.

5 Conclusion

In this paper, we propose a normalized generalized modularity criterion for binary and categorical data in the aim of co-clustering. We have studied its maximization. An efficient spectral procedure for optimization is presented, the experimental results obtained using different real world data sets show that our method works effectively for binary data. We obtain simultaneously row and columns clusters where each row cluster is characterized by a column cluster. Our method can be easily extended to more general spectral framework for combining multiples heterogeneous data sets for co-clustering.

Acknowledgment. This research was supported by the CLasSel ANR project ANR-08-EMER-002

References

1. White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: SDM, pp. 76–84 (2005)
2. Ng, A.Y., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Proc. of NIPS, vol.14 (2001)
3. Cheng, Y., Church, G.: Biclustering of expression data. In: 8th International Conference on Intelligent Systems for Molecular Biology, ISMB 2000, pp. 93–103 (2000)
4. Von Luxburg, U.: A Tutorial on Spectral Clustering, Technical Report at MPI Tuebingen (2006)
5. Huang, Z.: Extensions to the k-means algorithm for proposition clustering large data sets with categorical values. Data Mining and Knowledge Discovery 2, 283–304 (1998)
6. Newman, M., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E 69, 026113 (2004)
7. Ding, C., Xiaofeng, H., Hongyuan, Z., Horst, S.: Self-aggregation in scaled principal component space. Technical Report LBNL–49048. Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA, USA (2001)
8. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix trifactorizations for clustering. In: KDD 2006, Philadelphia, PA (2006)
9. Dhillon, I., Mallela, S., Modha, D.S.: Information-Theoretic Co-clustering. In: KDD 2003, pp. 89–98 (2003)
10. Dhillon, I.: Co-clustering documents and words using bipartite spectral graph partitioning. In: KDD 2001, pp. 269–274 (2001)
11. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
12. Govaert, G., Nadif, M.: Block clustering with Bernoulli mixture models: Comparison of different approaches. Computational Statistics and Data Analysis 52, 233–3245 (2008)
13. Golub, G.H., Van Loan, C.F.: Matrix Computations. John Hopkins Press (1999)
14. Marcotorchino, F.: Block seriation problems: A unified approach. In: Applied Stochastic Models and Data Analysis, vol. 3, p. 7391. Wiley (1987)
15. Niermann, S.: Optimizing the ordering of tables with Evolutionary Computation. Tha American Statistician 59(1), 41–46 (2005)

Co-clustering under Nonnegative Matrix Tri-Factorization

Lazhar Labiod and Mohamed Nadif

LIPADE, Université Paris Descartes
45 rue des Saints-Pères, 75006 Paris, France
{firstname.lastname}@parisdescartes.fr

Abstract. The nonnegative matrix tri-factorization (NMTF) approach has recently been shown to be useful and effective to tackle the co-clustering. In this work, we embed this problem in the NMF framework and we derive from the double k-means objective function a new formulation of the criterion. To optimize it, we develop two algorithms based on two multiplicative update rules. In addition we show that the double k-means is equivalent to algebraic problem of NMF under some suitable constraints. Numerical experiments on simulated and real datasets demonstrate the interest of our approach.

Keywords: nmf, double k-means, co-clustering.

1 Introduction

For datasets arising in text mining and bioinformatics where the data is represented in a very high dimensional space, clustering both dimensions of data matrix simultaneously is often more desirable than traditional one side clustering. Co-clustering which is a simultaneous clustering of rows and columns of data matrix consists in interlacing row clusterings with column clusterings at each iteration [1]; co-clustering exploits the duality between rows and columns which allows to effectively deal with high dimensional data. In [1], the authors proposed an information-theoretic co-clustering algorithm that presents a non-negative matrix as an empirical joint probability distribution of two discrete random variables and set co-clustering problem under an optimization problem in information theory. Model-based clustering techniques have also shown promising results in several co-clustering situations, the co-clustering of co-occurrence table has been treated by using latent block Poisson models [4]. The co-clustering implicitly performs an adaptive dimensionality reduction at each iteration, leading to better document clustering accuracy compared to one side clustering methods [1]. Co-clustering is also preferred when there is an association relationship between the data and the features (i.e., the columns and the rows) [2].

Even if the co-clustering problem is not the main objective of nonnegative factorization matrix (NMF), this approach has attracted many authors for data co-clustering and particularly for document clustering. Then, different algorithms based on nonnegative tri-factorization matrix are proposed. Given a nonnegative matrix A, they consist in seeking a 3-factor decomposition $\mathbf{U}S\mathbf{V}^T$ with all

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 709–717, 2011.
© Springer-Verlag Berlin Heidelberg 2011

factor matrices restricted to be nonnegative. The matrices $\mathbf{U}$ and $\mathbf{V}$ play the roles of row and column memberships. Each value of both matrices $\mathbf{U}$ and $\mathbf{V}$ corresponds to the degree in which a row or column belongs to a cluster. The matrix S makes it possible to absorb the scales of $\mathbf{U}$, $\mathbf{V}$ and A. All proposed algorithms are iterative, which can be differentiated by the update rules of the three matrices due to the chosen optimization method or the supplementary constraints imposed on the three matrices.

The approximation of A can be solved by an iterative alternating least-squares optimization procedure. For instance, the non-negative block value decomposition (NBVD) [7] offers a solution of this problem. At the convergence, and assuming that $\mathbf{U}A$ is normalized to $\mathbf{U}A\mathbf{X}$ ($\mathbf{X}$ is a diagonal matrix), the cluster labels of the columns, are deduced with $\mathbf{X}^{-1}\mathbf{V}^T$. We can also deduce the label cluster rows by working on A^T. Note that in NBVD only the nonnegativity of the three matrices is required. In [2] and [10] the authors emphasized the importance of the orthogonality constraint, they introduced it on $\mathbf{U}$ and $\mathbf{V}$ and proposed respectively ONM3F and ONMTF which can be differentiated by the update rules of the 3 factors. In a task of document clustering, NBVD, ONM3F and ONMTF were shown to work well. In this paper, we propose a new co-clustering framework based on NMF formulation. Contrary to previous approaches, we embed the co-clustering aim under the nonnegative factorization at the beginning. The key idea is that the latent block structure in a rectangular nonnegative data matrix is factorized into two factors rather than three factors, the row-coefficient matrix $\mathbf{R}$ and column-coefficient matrix $\mathbf{C}$ indicating respectively the degree in which a row and a column belong to a cluster. As our formulation arises from a reformulation of the double $k$means, we called it *double nonnegative matrix factorization* (DNMF), and ODNMF when the orthogonality constraints on $\mathbf{R}$ and $\mathbf{C}$ are required. Then, under this framework we develop a novel co-clustering algorithm for nonnegative data, which iteratively compute two factors based on two multiplicative update rules.

The rest of paper is organized as follows. Section 2 introduces notation and describes the general co-clustering model. Section 3 provides details on the new NMF framework for co-clustering. Sections 4 and 5 are devoted to propose update rules with different constraints on $\{\mathbf{R}, \mathbf{C}\}$ and to explore the connections with other NMF methods. In Section 6, we evaluate all discussed update rules for co-clustering from simulated and real data sets. Finally, the conclusion summarizes the advantages of our contribution.

2 Double *k*means

Given a data matrix $A = (a_{ij}) \in \mathcal{R}^{M \times N}$, the aim of co-clustering is to simultaneously cluster the rows and columns of A, so as to optimize the difference between $A = (a_{ij})$ and the clustered matrix revealing significant block structure. More formally, we seek to partition the set of rows $I = \{1, \ldots, M\}$ into K clusters $P = \{P_1, \ldots, P_K\}$ and the set of columns $J = \{1, \ldots, N\}$ into L clusters $Q = \{Q_1, \ldots, Q_L\}$. The two partitionings naturally induce clustering

index matrices $R = (r_{ik}) \in \mathcal{R}^{M\times K}$ and $C = (c_{j\ell}) \in \mathcal{R}^{N\times L}$, defined as binary classification matrices such as $\sum_{k=1}^{K} r_{ik} = 1$ $\sum_{\ell=1}^{L} c_{j\ell} = 1$. Specifically, we have $r_{ik} = 1$, if the row $\mathbf{a}_i \in P_k$, and 0 otherwise. The matrix $\mathbf{C}$ is defined similarly by $c_{j\ell} = 1$, if the column $\mathbf{a}_j \in Q_\ell$, and 0 otherwise. Thanks to r_{ik} and $c_{j\ell}$, a submatrix or block $A_{k\ell}$ is therefore defined by $\{(a_{ij})|r_{ik}c_{j\ell}a_{ij} = 1\}$. On the other hand, we note $S = (s_{k\ell}) \in \mathcal{R}^{K\times L}$ a reduced matrix specifying the cluster representation.

In the following Upper-case letters generally denote matrices while lower-case boldfaced letters denote vectors, and not boldfaced denote scaler. The norm $||.||$ of matrix denotes the Frobenius norm, i.e., $||A||^2 = \sum_{i,j} A_{ij}^2$ while the symbol $|.|$ denotes the cardinality of cluster. The superscript T denotes matrix transposition while $\odot$ represents Hadamard product. Finally to simplify the notation, the sums relating to rows, columns or clusters will be subscripted respectively by the letters i, j, k or ℓ without indicating the limits of variation, which are implicit. The detection of homogeneous blocks in A can be reached by looking for the three matrices R, C and S minimizing the total squared residue measure

$$\mathcal{J}(A, RSC^T) = ||A - RSC^T||^2 \tag{1}$$

The term RSC^T characterizes the information of A that can be described by the cluster structures. The clustering problem can be formulated as a matrix approximation problem where the clustering aim is to minimize the approximation error between the original data A and the reconstructed matrix based on the cluster structures. Note that this matricial formulation can take the following form $\mathcal{J}(A, RSC^T) = \sum_{i,j,k,\ell} r_{ik}c_{j\ell}(a_{ij} - s_{k\ell})^2$. With fixed P_k and Q_ℓ, it is easy to check that the optimum S is obtained by $s_{k\ell} = \frac{\sum_{i,j,k,\ell} r_{ik}c_{j\ell}a_{ij}}{r_k c_\ell}$ where, $r_k = |P_k|$ and $c_\ell = |Q_\ell|$. In other words, each $s_{k\ell}$ is the centroid of block $A_{k\ell}$.

The approximation of A can be solved by an iterative alternating least-squares optimization procedure. When A is not necessarily non negative, different algorithms have been proposed to minimize this criterion (see for instance, [6]). These algorithms are equivalent and consist in using the principle of a double $k$means. Simplicity and scalability are the advantages of this algorithm. A version based on update rules is present in Algorithm 1. When $A \geq 0$, different algorithms arising from the non-negative matrix factorization approach offer a solution of the problem (1) with only the nonnegativity constraints on R, S and

Algorithm 1. Double $k$means

Input: data A, number of clusters K and L
Output: data R, S and C
Initialize: R and C
Compute: S, $s_{k\ell} = \sum_{i,j} \frac{(r_{ik}c_{j\ell})a_{ij}}{r_k c_\ell}$
repeat
-Update R, using $r_{ik} = 1$ if $k = \text{argmin}_{1\leq k'\leq K} \sum_{j,\ell} c_{j\ell}(a_{ij} - s_{k'\ell})^2$
-Update C, using $c_{j\ell} = 1$ if $\ell = \text{argmin}_{1\leq \ell'\leq L} \sum_{i,k} r_{j\ell}(a_{ij} - s_{k\ell'})^2$
-Update S
until *noChange* of $||A - RSC^T||^2$

C (R and C are not necessarily binary) [7], [6] and [8]. Next, we see how we can convert the double $k$means criterion to an optimization problem under the NMF approach.

3 NMF Framework for Co-clustering

By considering the double $k$means as a lower rank matrix factorization with constraints rather than a clustering method, we can formulate constraints to impose on NMF formulation. As shown above, in double $k$means clustering the objective function to be minimized is the sum of squared distance from row and column data to their centroid.

Let $D_r^{-1} \in \mathcal{R}^{K\times K}$ and $D_c^{-1} \in \mathcal{R}^{L\times L}$ be diagonals matrices defined as follow $D_r^{-1} = Diag(r_1^{-1}, \ldots, r_K^{-1})$ and $D_c^{-1} = Diag(c_1^{-1}, \ldots, c_L^{-1})$. Using the matrices D_r, D_c, A, R and C, the matrix summary S can be expressed as $D_r^{-1}R^TACD_c^{-1}$. Plugging S into the objective function equation (1), the expression to optimize becomes $||A - R(D_r^{-1}R^TACD_c^{-1})C^T||^2 = ||A - \mathbf{R}\mathbf{R}^TA\mathbf{C}\mathbf{C}^T||^2$, where $\mathbf{R} = RD_r^{-0.5}$ and $\mathbf{C} = CD_c^{-0.5}$. Note that this formulation holds even if A is not nonnegative, i.e., A has mixed signs entries. On the other hand, it is easy to check that the approximation $\mathbf{R}\mathbf{R}^TA\mathbf{C}\mathbf{C}^T$ of A is formed by the same value in each block $A_{k\ell}$. Specifically, the matrix $\mathbf{R}^TA\mathbf{C}$, equal to S, plays the role of a summary of A and absorbs the different scales of A, $\mathbf{R}$ and $\mathbf{C}$. Finally the matrices $\mathbf{R}\mathbf{R}^TA$, $A\mathbf{C}\mathbf{C}^T$ give respectively the row and column clusters mean vectors. Next, we propose an example to illustrate the roles of different matrices. Let A be a (4×5) matrix $\begin{pmatrix} 3 & 3 & 3 & 8 & 7 \\ 3 & 4 & 5 & 7 & 6 \\ 1 & 0 & 2 & 5 & 5 \\ 0 & 1 & 2 & 5 & 5 \end{pmatrix}$. Let R and C be the binary classification data matrices of A: $R = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$ and $C = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$. The matrices $\mathbf{R} = RD_r^{-0.5}$ and $\mathbf{C} = CD_c^{-0.5}$ are defined as follows $\mathbf{R} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} \end{pmatrix}$ and $\mathbf{C} = \begin{pmatrix} \frac{1}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{3}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} \end{pmatrix}$. The rows of $A\mathbf{C}\mathbf{C}^T$ are the row mean vectors $A\mathbf{C}\mathbf{C}^T = \left(\begin{array}{ccc|cc} 3 & 3 & 3 & 7.5 & 7.5 \\ 4 & 4 & 4 & 6.5 & 6.5 \\ \hline 1 & 1 & 1 & 5 & 5 \\ 1 & 1 & 1 & 5 & 5 \end{array}\right)$. The columns of $\mathbf{R}\mathbf{R}^TA$ are the column mean vectors $\mathbf{R}\mathbf{R}^TA = \left(\begin{array}{ccc|cc} 3 & 3.5 & 4 & 7.5 & 6.5 \\ 3 & 3.5 & 4 & 7.5 & 6.5 \\ \hline 0.5 & 0.5 & 2 & 5 & 5 \\ 0.5 & 0.5 & 2 & 5 & 5 \end{array}\right)$. The approximation of A is defined by $\mathbf{R}\mathbf{R}^TA\mathbf{C}\mathbf{C}^T = \left(\begin{array}{ccc|cc} 3.5 & 3.5 & 3.5 & 7 & 7 \\ 3.5 & 3.5 & 3.5 & 7 & 7 \\ \hline 1 & 1 & 1 & 5 & 5 \\ 1 & 1 & 1 & 5 & 5 \end{array}\right)$. The centroid is $\mathbf{R}^TA\mathbf{C} = \left(\begin{array}{c|c} 8.57 & 14 \\ \hline 2.44 & 10 \end{array}\right)$ and we have $||A - \mathbf{R}\mathbf{R}^TA\mathbf{C}\mathbf{C}^T||^2 = 3.08$.

Setting double $k$means in the NMF approach, the problem of co-clustering can be reformulated as the seek of $\mathbf{R}$ and $\mathbf{C}$ minimizing $||A - \mathbf{R}\mathbf{R}^TA\mathbf{C}\mathbf{C}^T||^2$. The compute of $\mathbf{R}$ and $\mathbf{C}$ is difficult and requires an iterative algorithm. But, many properties are satisfied by $\mathbf{R}$ and $\mathbf{C}$ can be easily proved and illustrated by the previous example. Then and contrary to double $k$means, in the following we propose a continuous optimization under suitable constraints generated by the properties of $\mathbf{R}$ and $\mathbf{C}$.

In the rest of this paper, we shall only focus on the case where $A \geq 0$. In this section, we tackle the problems of double NMF without constraint of orthogonality which we called DNMF, and with constraint of orthogonality called ODNMF, we derive the multiplicative update rules for DNMF using the Karush-Kuhn-Tucker (KKT) conditions. For ODMNF, the updating rules will be derived by exploiting the true gradients on Stiefel manifolds [3].

4 DNMF Formulation

In this subsection we consider only the nonnegativity constraint, the objective function becomes $\text{argmin}_{\mathbf{R},\mathbf{C}\geq 0} ||A - \mathbf{R}\mathbf{R}^T A\mathbf{C}\mathbf{C}^T||^2$. We aim to optimize the quadratic form above with nonnegativity constraint on both $\mathbf{R}$ and $\mathbf{C}$. We follow the standard optimization theory and derive the KKT conditions. To find the minima, we introduce the Lagrangian function defined by $\mathcal{L} = ||A - \mathbf{R}\mathbf{R}^T A\mathbf{C}\mathbf{C}^T||^2 - Trace(\Lambda\mathbf{R}^T) - Trace(\Gamma\mathbf{C}^T)$, where the lagrangian multiplier matrices Λ and Γ are to enforce nonnegative constraints respectively on $\mathbf{R}$ and $\mathbf{C}$. The KKT conditions require: $\frac{\partial||A-\mathbf{R}\mathbf{R}^T A\mathbf{C}\mathbf{C}^T||^2}{\partial\mathbf{R}} = \Lambda$ and $\frac{\partial||A-\mathbf{R}\mathbf{R}^T A\mathbf{C}\mathbf{C}^T||^2}{\partial\mathbf{C}} = \Gamma$ as optimality conditions and $\Lambda \odot \mathbf{R} = 0$ and $\Gamma \odot \mathbf{C} = 0$ as complementarity slackness conditions. This leads to $[2AX_{\mathbf{C}}^T\mathbf{R} - (\mathbf{R}\mathbf{R}^T X_{\mathbf{C}} X_{\mathbf{C}}^T\mathbf{R} + X_{\mathbf{C}} X_{\mathbf{C}}^T\mathbf{R}\mathbf{R}^T\mathbf{R})] \odot \mathbf{R} = 0$ and $[2X_{\mathbf{R}}^T A\mathbf{C} - (\mathbf{C}\mathbf{C}^T X_{\mathbf{R}} X_{\mathbf{R}}^T\mathbf{C} + X_{\mathbf{R}} X_{\mathbf{R}}^T\mathbf{C}\mathbf{C}^T\mathbf{C})] \odot \mathbf{C} = 0$ with, $X_{\mathbf{C}} = A\mathbf{C}\mathbf{C}^T$ and $X_{\mathbf{R}} = \mathbf{R}\mathbf{R}^T A$. That leads to the following multiplicative update rules:

$$\mathbf{R} \leftarrow \mathbf{R} \odot \frac{2AX_{\mathbf{C}}^T\mathbf{R}}{\mathbf{R}\mathbf{R}^T X_{\mathbf{C}} X_{\mathbf{C}}^T\mathbf{R} + X_{\mathbf{C}} X_{\mathbf{C}}^T\mathbf{R}\mathbf{R}^T\mathbf{R}}, \tag{2}$$

$$\mathbf{C} \leftarrow \mathbf{C} \odot \frac{2X_{\mathbf{R}}^T A\mathbf{C}}{\mathbf{C}\mathbf{C}^T X_{\mathbf{R}} X_{\mathbf{R}}^T\mathbf{C} + X_{\mathbf{R}} X_{\mathbf{R}}^T\mathbf{C}\mathbf{C}^T\mathbf{C}}. \tag{3}$$

We derive an algorithm to compute the nonnegative relaxation. The algorithm has classical steps of NMF. Given an existing solution or an initial guess, we iteratively improve the solution by updating the factors with the rules (2) and (3). To prove the convergence of our algorithm, following Lee and Seung [5], we can easily use the similar concept of auxiliary function approach to achieve this goal. Hereafter, we analyze the relationship among different types of NMF.

First, we emphasize the sens of our formulation which can be viewed as a non nonnegative matrix tri-factorization method. Indeed, it is equivalent to nonnegative block value decomposition (NBVD) with respect to constraint S to be equal to $\mathbf{R}^T A\mathbf{C}$, then we have to consider, $\text{argmin}_{\mathbf{R},\mathbf{C}\geq 0} ||A - \mathbf{R}S\mathbf{C}^T||^2$ s.t. $S = \mathbf{R}^T A\mathbf{C}$. On the other hand, if we consider the additional constraints $\mathbf{R}^T\mathbf{R} = I_K$ and $\mathbf{C}^T\mathbf{C} = I_L$, DNMF becomes equivalent to orthogonal nonnegative tri-factorization (ONM3F) proposed in [10].

In the same way, we analyze the relationship among different types of NMF. As for the double $k$means objective function, we can show $k$means as a constrained NMF problem. Indeed, taking $S = A^T R D_r^{-1}$, the criterion to be optimized can be written as $\mathcal{J}_{kmeans} = \sum_k \sum_{i|r_{ik}=1} ||\mathbf{a}_i - \mathbf{s}_k||^2 = ||A - RS^T||^2$. Plugging S into $\mathcal{J}_{kmeans}$, we obtain $||A - RD_r^{-1}R^T A||^2 = ||A - \mathbf{R}\mathbf{R}^T A||^2$. With respect

to the nonnegativity of $\mathbf{R}$ this minimization leads to the following update rule $\mathbf{R} \leftarrow \mathbf{R} \odot \frac{2AA^T\mathbf{R}}{\mathbf{R}\mathbf{R}^TAA^T\mathbf{R}+AA^T\mathbf{R}\mathbf{R}^T\mathbf{R}}$. This multiplicative update rule is similar to that obtained by using the projective nonnegative matrix factorization [9].

5 ODNMF Formulation

To derive the multiplicative update rules with respect to orthogonality constraints on $\mathbf{R}$ and $\mathbf{C}$, we compute the true gradients (or natural gradient) on Stiefel manifolds. With the same notations used in [10], the true gradients of the objective function $\mathcal{E} = ||A - \mathbf{R}\mathbf{R}^TA\mathbf{C}\mathbf{C}^T||^2$ are computed as

$$\tilde{\nabla}_{\mathbf{R}}\mathcal{E} = \nabla_{\mathbf{R}}\mathcal{E} - \mathbf{R}[\nabla_{\mathbf{R}}\mathcal{E}]^T\mathbf{R} = [\tilde{\nabla}_{\mathbf{R}}\mathcal{E}]^+ - [\tilde{\nabla}_{\mathbf{R}}\mathcal{E}]^-$$

$$\tilde{\nabla}_{\mathbf{C}}\mathcal{E} = \nabla_{\mathbf{C}}\mathcal{E} - \mathbf{C}[\nabla_{\mathbf{C}}\mathcal{E}]^T\mathbf{C} = [\tilde{\nabla}_{\mathbf{C}}\mathcal{E}]^+ - [\tilde{\nabla}_{\mathbf{C}}\mathcal{E}]^-,$$

where $[\tilde{\nabla}_{\mathbf{R}}\mathcal{E}]^+$, $[\tilde{\nabla}_{\mathbf{R}}\mathcal{E}]^-$, $[\tilde{\nabla}_{\mathbf{C}}\mathcal{E}]^+$ and $[\tilde{\nabla}_{\mathbf{C}}\mathcal{E}]^-$ are positive. Then the updating rules take the following form

$$\mathbf{R} = \mathbf{R} \odot \frac{[\tilde{\nabla}_{\mathbf{R}}\mathcal{E}]^-}{[\tilde{\nabla}_{\mathbf{R}}\mathcal{E}]^+} \text{ and } \mathbf{C} = \mathbf{C} \odot \frac{[\tilde{\nabla}_{\mathbf{C}}\mathcal{E}]^-}{[\tilde{\nabla}_{\mathbf{C}}\mathcal{E}]^+}. \tag{4}$$

The true gradients on Stiefel manifolds, $\{\mathbf{R}|\mathbf{R}^T\mathbf{R} = I_K\}$, $\{\mathbf{C}|\mathbf{C}^T\mathbf{C} = I_L\}$, are then $\tilde{\nabla}_{\mathbf{R}}\mathcal{E} = \mathbf{R}\mathbf{R}^TA\mathbf{C}\mathbf{C}^TA^T\mathbf{R} - A\mathbf{C}\mathbf{C}^TA^T\mathbf{R}$ and $\tilde{\nabla}_{\mathbf{C}}\mathcal{E} = \mathbf{C}\mathbf{C}^TA^T\mathbf{R}\mathbf{R}^TA\mathbf{C} - A^T\mathbf{R}\mathbf{R}^TA\mathbf{C}$. Using the relations of (4) with the gradient calculations, we obtain the following multiplicative updating rules

$$\mathbf{R} \leftarrow \mathbf{R} \odot \frac{A\mathbf{C}\mathbf{C}^TA^T\mathbf{R}}{\mathbf{R}\mathbf{R}^TA\mathbf{C}\mathbf{C}^TA^T\mathbf{R}}, \tag{5}$$

$$\mathbf{C} \leftarrow \mathbf{C} \odot \frac{A^T\mathbf{R}\mathbf{R}^TA\mathbf{C}}{\mathbf{C}\mathbf{C}^TA^T\mathbf{R}\mathbf{R}^TA\mathbf{C}}. \tag{6}$$

We note that plugging $S = \mathbf{R}^TA\mathbf{C}$ in (5,6) and in those obtained by ONMTF (reported in table 1), we have exactly the same update rules. However and contrary to NMTF or ONMTF only two matrices need to be solved rather than three. Further, due to the double product $\mathbf{R}\mathbf{R}^T$ and $\mathbf{C}\mathbf{C}^T$ in the updates rules, our approach provides more sparse matrices, which facilitate the interpretation of the resulting factors.

6 Numerical Experiments

In this section we present a set of experiments on synthetic and real world dyadic data to validate the effectiveness of our proposed algorithms for co-clustering. We compare the performance of the proposed DNMF and ODNMF methods with other competitive algorithms, NBVD, ONM3F, ONMTF. For all the algorithms we use the same datasets. As we focus on document clustering, we use a normalization TF-IDF. Then, we randomly initialize the matrices updates over

500 iterations, we calculate the clustering accuracy, normalized mutual information and the loss objective function (goodness of fit) at each iteration. Finally, we report Acc and NMI corresponding to the smallest goodness. To avoid confusion with $\mathbf{R}$ and $\mathbf{C}$ computed in DNMF (formulas (2,3)) and ODNMF (formulas (5,6)), we prefer to denote the row and column coefficients matrices by $\mathbf{U}$ and $\mathbf{V}$ corresponding to the classical approximation in tri-factorization $A \approx \mathbf{U}S\mathbf{V}^T$. The update rules of $\mathbf{U}$, $\mathbf{V}$ and the third factor S by NBVD, ONM3F and ONMTF are reported in Table 1.

Table 1. Algorithms and update rules

Factors	NBVD	ONM3F	ONMTF
$\mathbf{U} \leftarrow$	$\mathbf{U} \odot \frac{A\mathbf{V}S^T}{\mathbf{U}S\mathbf{V}^T\mathbf{V}S^T}$	$\mathbf{U} \odot (\frac{A\mathbf{V}S^T}{\mathbf{U}\mathbf{U}^TA\mathbf{V}S^T})^{-\frac{1}{2}}$	$\mathbf{U} \odot (\frac{A\mathbf{V}S^T}{\mathbf{U}S\mathbf{V}^TA^T\mathbf{U}})$
$\mathbf{V} \leftarrow$	$\mathbf{V} \odot \frac{A^T\mathbf{U}S}{\mathbf{V}S^T\mathbf{U}^T\mathbf{U}S}$	$\mathbf{V} \odot (\frac{A^T\mathbf{U}S}{\mathbf{V}\mathbf{V}^TA^T\mathbf{U}S})^{-\frac{1}{2}}$	$\mathbf{V} \odot (\frac{A^t\mathbf{U}S}{\mathbf{V}S^T\mathbf{U}^TA\mathbf{V}})$
$S \leftarrow$	$S \odot \frac{\mathbf{U}^TA\mathbf{V}}{\mathbf{U}^T\mathbf{U}S\mathbf{V}^T\mathbf{V}}$	$S \odot (\frac{\mathbf{U}^TA\mathbf{V}}{\mathbf{U}^T\mathbf{U}S\mathbf{V}^T\mathbf{V}})^{-\frac{1}{2}}$	$S \odot \frac{\mathbf{U}^TA\mathbf{V}}{\mathbf{U}^T\mathbf{U}S\mathbf{V}^T\mathbf{V}}$

To evaluate the clustering results, we adopt the clustering accuracy and normalized mutual information performance measures. We only focus on the quality of row clustering. Clustering Accuracy noted Acc discovers the one-to-one relationship between clusters and classes and measures the extent to which each cluster contained data points from the corresponding class; it is defined as follows: $Acc = \frac{1}{N} max[\sum_{\mathcal{C}_k,\mathcal{L}_m} T(\mathcal{C}_k, \mathcal{L}_m)]$, where $\mathcal{C}_k$ is the kth cluster in the final results, and $\mathcal{L}_m$ is the true mth class. $T(\mathcal{C}_k, \mathcal{L}_m)$ is the number of entities which belong to class m and are assigned to cluster k. Accuracy computes the maximum sum of $T(\mathcal{C}_k, \mathcal{L}_m)$ for all pairs of clusters and classes, and these pairs have no overlaps. The greater clustering accuracy means, the better clustering performance. The second measure employed is the Normalized Mutual Information (NMI); it is estimated by $\frac{\sum_{k,\ell} N_{k,\ell} \log \frac{N_{k,\ell}}{N_k \hat{N}_\ell}}{\sqrt{(\sum_k N_k \log \frac{N_k}{N})(\sum_\ell \hat{N}_k \log \frac{\hat{N}_\ell}{N})}}$, where N_k denotes the number of data contained in the cluster $\mathcal{C}_k (1 \leq k \leq K)$, $\hat{N}_\ell$ is the number of data belonging to the class $\mathcal{L}_k (1 \leq k \leq K)$, and $N_{k,\ell}$ denotes the number of data that are in the intersection between the cluster $\mathcal{C}_k$ and the class $\mathcal{L}_\ell$. The larger the NMI, the better the clustering result.

6.1 Synthetic Data

To generate nonnegative data, we propose to use a latent block model [4]. The authors have proposed the model defined by the following pdf $f(A; \boldsymbol{\theta}) = \sum_{(R,C)\in\mathcal{R}\times\mathcal{W}} p(R; \boldsymbol{\theta}) p(C; \boldsymbol{\theta}) f(A|R, C; \boldsymbol{\theta})$, where $\mathcal{R}$ and $\mathcal{C}$ denote the sets of all possible assignments R of I and C of J. Now, as in latent class analysis, the $M \times N$ random variables generating the observed a_{ij} cells are assumed to be independent once R and C are fixed; then it is written $f(A|R, C; \boldsymbol{\theta}) = \prod_{i,j,k,\ell} \varphi(a_{ij}; \alpha_{k\ell})^{r_{ik}c_{j\ell}}$, where $\varphi(.; \alpha_{k\ell})$ is a pdf defined on the real set and $\alpha_{k\ell}$ an unknown parameter. The parameter $\boldsymbol{\theta}$ is formed by $(\boldsymbol{\alpha}, \mathbf{p}, \mathbf{q})$ where $\boldsymbol{\alpha} = (\alpha_{11}, \ldots, \alpha_{KL})$, $\mathbf{p} = (p_1, \ldots, p_K)$ and $\mathbf{q} = (q_1, \ldots, q_L)$ are the vectors

of probabilities p_k and q_ℓ that a row and a column belong to the kth component and to the ℓth component respectively. For co-occurrence table, in [4] the authors have proposed a Poisson latent block model which assumes that for each block $k\ell$ the values a_{ij} are distributed according the Poisson distribution $P(\mu_i\nu_j\alpha_{k\ell})$ where the Poisson parameter is split into μ_i and ν_j the effects of the row i and the column j and $\alpha_{k\ell}$ the effect of the block $k\ell$ consisting of the cells belonging to the kth row cluster and ℓth column cluster. Then φ for block $k\ell$ is defined as follows $\frac{exp(-\mu_i\nu_j\alpha_{k\ell})(\mu_i\nu_j\alpha_{k\ell})^{a_{ij}}}{a_{ij}!}$, where the effects $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_M)$ and $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_N)$ are assumed equal to the marginal totals. Note that as $a_{ij} \in \mathbb{N}$, the parameters to estimated $\alpha_{k\ell}$ belong to $\mathbb{R}^+$, and the independence of the features of R and C provide the probabilities $p(R;\boldsymbol{\theta}) = \prod_{i,k} p_k^{r_{ik}}$ and $p(C;\boldsymbol{\theta}) = \prod_{j,\ell} q_\ell^{c_{j\ell}}$. From our numerical experiments, we considered different situations corresponding to various levels of overlap degrees. We simulated 500×500 non-negative data arising from a 3×3-component mixture model with $\mathbf{p} = (0.3, 0.3, 0.4)$ and $\mathbf{q} = (0.25, 0.25, 0.5)$. The parameters of $\boldsymbol{\alpha}$ are chosen to yield different degrees of overlap. The main points arising from experiments in *Acc* and NMI terms are the following. ODNMF, ONM3F and ONMTF outperform DNMF and NBVD; then we note the importance of the orthogonality constraint. ODNMF appears more preferable to ONM3F and ONMTF while DNMF is often superior to NBVD (Table 3).

Table 2. Clustering performance evaluation with $K = 3$ and $L = 3$

degree of overlap	performance measure	DNMF	ODNMF	ONM3F	ONMTF	NBVD
3%	Acc	92.66	**96.33**	87.16	95.41	87.16
	NMI	83.30	**89.46**	75.91	88.81	70.46
10%	Acc	61.03	**67.65**	65.44	61.03	59.56
	NMI	41.78	43.92	41.32	**46.06**	33.37
15%	Acc	59.15	**66.90**	65.49	65.49	64.08
	NMI	37.28	43.32	**48.77**	46.40	43.29
24%	Acc	53.07	**59.78**	56.98	55.87	52.51
	NMI	27.01	35.16	**36.00**	35.30	28.41
37%	Acc	53.00	**54.00**	51.00	52.00	48.00
	NMI	18.11	**24.66**	20.91	16.02	10.99

6.2 Real Datasets

We used three datasets for document clustering. Classic30 is an extract of Classic3 [1] which counts three classes denoted Medline, Cisia, Cranfield as their original database source. It consists of 30 random documents described by 1000 words. Classic150 consists of 150 random documents described by 3652 words. Finally, NG2 is a subset of 20-Newsgroup data NG20, it is composed by 500 documents concerning talk.politics.mideast and talk.politics.misc. We have used these datasets with $M << N$ in order to prove the benefit of ODNMF, ONM3F and ONMTF. The normalized cut weighting defined by [8] is applied to data before applying clustering algorithms. In our experiments, we have taken K equal to the true numbers of document clusters while L is taken equal to 10 for all datasets due to the size of the set of words. Some investigations are underway in order to better choose the parameter L. The results on real data confirm the performance recorded in synthetic data; ODNMF has a good behavior.

Table 3. Performance evaluation on Classic30 ($K = 3, L = 10$), Classic150 ($K = 3, L = 10$) and NG2 with ($K = 2, L = 10$)

dataset	performance measure	DNMF	ODNMF	ONM3F	ONMTF	NBVD
Classic30	Acc	96.67	**100**	**100**	**100**	96.67
	NMI	89.97	**100**	**100**	**100**	89.97
Classic150	Acc	98.66	98.66	**99.33**	98.66	98.66
	NMI	94.04	94.04	**97.02**	94.04	94.04
NG2	Acc	77.6	**86.2**	74.6	74.2	77.4
	NMI	19.03	**43.47**	18.27	16.03	23.31

7 Conclusion

Starting from the double $k$means objective function, we have proposed a new co-clustering framework based on NMF formulation. Contrary to previous approaches, we embedded the co-clustering aim under the nonnegative factorization, at the beginning. We have shown that the double $k$means can be formulated as an optimization of the objective function under a set of suitable constraints generated by the properties of two factors $\mathbf{R}$ and $\mathbf{C}$. We have seen that different co-clustering approaches proposed in the literature can be derived from this new framework and contrary to NMTF, with our approach only two matrices need to be solved rather than three.

Acknowledgment. This research was supported by the CLasSel ANR project ANR-08-EMER-002.

References

1. Dhillon, I., Mallela, S., Modha, D.S.: Information-theoretic coclustering. In: Proceedings of KDD 2003, pp. 89–98 (September 2003)
2. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix tri-factorizations for clustering. In: Proceedings of KDD 2006, Philadelphia, PA, pp. 635–640 (September 2006)
3. Edelman, A., Arias, T., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM Journal of Matrix Analysis and Application 20(2), 303–353 (1998)
4. Govaert, G., Nadif, M.: Latent block model for contingency table. Communications in Statistics, Theory and Methods 39, 416–425 (2010)
5. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems NIPS, vol.13, pp. 303–353. MIT Press (2001)
6. Li, T., Ma, S.: Iterative feature and data clustering. In: International Conference on Data Mining (SDM), pp. 536–543. SIAM (September 2004)
7. Long, B., Zhang, Z., Yu, P.S.: Co-clustering by value decomposition. In: Procedings of KDD 2005, pp. 635–640 (September 2005)
8. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: Proceedings of the ACM SIGIR 2003, Toronto, Canada, pp. 267–273 (September 2003)
9. Yang, Z., Yuan, Z., Laaksonen, J.: Projective non-negative matrix factorization with application to facial image processing. International Journal on Pattern Regognition and Artificial Intelligence 21, 1353–1362 (2007)
10. Yoo, J., Choi, S.: Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds. Information Processing and Management 46(5), 559–570 (2010)

SPAN: A Neuron for Precise-Time Spike Pattern Association

Ammar Mohemmed[1], Stefan Schliebs[1], and Nikola Kasabov[1,2]

[1] Knowledge Engineering Discovery Research Institute,
350 Queen Street, Auckland 1010, New Zealand
{amohemme,sschlieb,nkasabov}@aut.ac.nz
http://www.kedri.info
[2] Institute for Neuroinformatics, ETH and University of Zurich

Abstract. In this paper we propose SPAN, a LIF spiking neuron that is capable of learning input-output spike pattern association using a novel learning algorithm. The main idea of SPAN is transforming the spike trains into analog signals where computing the error can be done easily. As demonstrated in an experimental analysis, the proposed method is both simple and efficient achieving reliable training results even in the context of noise.

Keywords: Spiking Neural Networks, Supervised Learning, Nuerocomputing, Spatiotemporal pattern recognition.

1 Introduction

The neurons in the mammalian brain communicate with each other by exchanging short electrical pulses called spikes. Representing information in the form of spike sequences appears to be a powerful concept that has led to the development of Spiking Neural Networks (SNN) in which the neural information processing of the brain is mimicked. We refer to [4] for a comprehensive standard text on the material. Based on biological evidence it was shown that information can be principally encoded by the precise timing of spikes [1]. Due to the inheritance of the time in its functionality, SNN have attracted a growing interest especially in the context of spatio-temporal information processing [6]. Many of these applications involves pre-designed tasks where the system is trained in a supervised fashion. Only recently, a number of supervised learning algorithms for SNN are proposed.

One of the first supervised learning methods for SNN that is based on precise spike time encoding is SpikeProb [2]. It is a gradient descent approach that adjusts the synaptic weights in order to emit a single spike at specified time. The timing of the output spike encodes a specific information, *e.g.* the class label of the presented input sample. Using SpikeProp, the SNN can not be trained to emit more than one spike.

The so-called Tempotron is a neuron capable of learning whether to fire or not to fire in response to a specific spatio-temporal input stimulus [7]. Consequently,

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 718–725, 2011.
© Springer-Verlag Berlin Heidelberg 2011

the method is more suitable for binary classification problems. However, the time of the input individual spikes is important but the time of the output spike does not carry any information. Therefore, Tempotron is not capable to learn specific target output spike trains.

A Hebbian based supervised learning algorithm called Remote Supervised Method (ReSuMe) was proposed in [10]. ReSuMe, similar to spike time dependent plasticity (STDP) [8], is based on a learning window concept. Using a teaching signal specific desired output is imposed to the output neuron. With this method, a neuron is able to produce a spike train precisely matching the desired spike train. It was shown that in combination with the Liquid State Machine (LSM) [9], several temporal spike tasks can be performed including classification and random mapping from any input spike train to any output spike train.

Recently, a method called Chronotron was proposed for learning the mapping of a precisely timed input/output spike trains [3]. Two versions of learning rules were described. The first denoted by E-Learning is based on optimizing the parameters of the spiking neurons, the synaptic weights, to minimize the error between desired spike pattern and actual one. The error is measured using the Victor-Purpura spike distance metric [12]. This metric produces discontinuities in the error landscape due to the addition and removal of spikes that must be overcome through approximation. E-Learning was compared with ReSuMe on a temporal classification tasks and its better performance in terms of the number of spike patterns that can be classified was shown. The other version is called I-Learning which is more biologically plausible but less efficient.

In this paper we propose a new supervised learning algorithm for precise-time spike pattern association, which we call it SPAN for Spike Pattern Association Neuron. The algorithm, which SPAN is based on, is algorithmically less complex compared to the ones mentioned above but at the same time efficient in solving spatio-temporal classification tasks. The method is based on the well known Widrow-Hoff or Delta rule. However, applying this rule directly for spike trains is not possible because subtracting or multiplying spike trains is meaningless. We address this issue by convolving the spike train with a kernel function to convert the spike train into a continuous-valued signal. Thus, subtraction and multiplication of the convolved spike trains are possible and hence it is easy to compute the error between a desired and actual spike trains and hence to adjust the weights accordingly.

2 Learning Method

In this section, we describe the neural and synaptic model used in this study, followed by a detailed description of the proposed learning algorithm.

2.1 Neural and Synaptic Model

The Leaky Integrate-and-Fire (LIF) neuron is arguably the best known model for simulating spiking networks [4]. It is based on the idea of an electrical circuit

containing a capacitor with capacitance C and a resistor with a resistance R, where both C and R are assumed to be constant. The dynamics of a neuron i are then described by the following differential equation:

$$\tau_m \frac{du_i}{dt} = -u_i(t) + R\ I_i^{\text{syn}}(t) \tag{1}$$

The constant $\tau_m = RC$ is called the membrane time constant of the neuron. Whenever the membrane potential u_i crosses a threshold ϑ from below, the neuron fires a spike and its potential is reset to a reset potential u_{reset}. Following [4], we define

$$t_i^{(f)} : u_i(t^{(f)}) = \vartheta, f \in \{0, \ldots, k-1\} \tag{2}$$

as the firing times of neuron i where k is the number of spikes emitted by neuron i. It is noteworthy that the shape of the spike itself is not explicitly described in the traditional LIF model. Only the firing times are considered to be relevant.

The synaptic current I_i^{syn} of neuron i is modeled using an α-kernel:

$$I_i^{\text{syn}}(t) = \sum_j w_{ij} \sum_f \alpha(t - t_j^{(f)}) \tag{3}$$

where $w_{ij} \in \mathbb{R}$ is the synaptic weight describing the strength of the connection between neuron i and its pre-synaptic neuron j. The α-kernel itself is defined as

$$\alpha(t) = e\ \tau_s^{-1}\ t\ e^{-t/\tau_s} \Theta(t) \tag{4}$$

where $\Theta(t)$ refers to the Heaviside function.

2.2 Learning

Similar to other supervised training algorithms, the synaptic weights of the network are adjusted iteratively in order to produce a desired spike pattern in response to a specific input spike pattern. We start with the common Widrow-Hoff rule for modifying the weight of a synapse i:

$$\Delta w_i^{\text{WH}} = \lambda x_i\ (y_d - y_{\text{out}}) \tag{5}$$

where $\lambda \in \mathbb{R}$ is a real-valued positive learning rate, x_i is the input transferred through synapse i, and y_d and y_{out} refer to the desired and the actual network output, respectively.

This equation is not applicable directly for spike trains as in SNN. In order to define the distance between spike trains, we convolve each spike sequence with a kernel function. This is similar to the bin-less distance metric used to compare spike trains [11]. In this study, we use an α-kernel, however other kernels could be also applied. The convolved input spike train $t_{\text{i}}^{(f)}$ is defined as

$$x_i(t) = \sum_f \alpha(t - t_i^{(f)}) \tag{6}$$

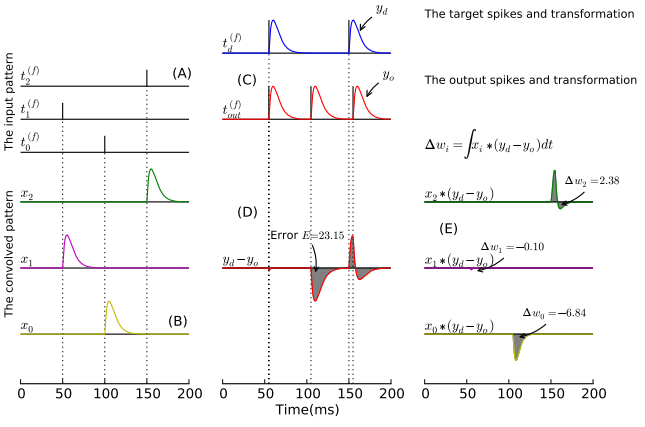


Fig. 1. Illustration of the proposed learning rule SPAN. See text for detailed explanations of the figure.

where $\alpha(t)$ refers to the α-function defined in Equation 4. Representing spikes as a function allows us to define the difference between spike sequences as the difference between their representing functions. Similar to the neural input, we define the desired and actual outputs of a neuron:

$$y_d(t) = \sum_f \alpha(t - t_d^{(f)}) \tag{7}$$

$$y_{\text{out}}(t) = \sum_f \alpha(t - t_{\text{out}}^{(f)}) \tag{8}$$

As a consequence of the spike representation defined above, Δw_i^{WH} itself is a function over time. By integrating Δw_i^{WH} we obtain a scalar Δw_i that is used to update the weight of synapse i:

$$\Delta w_i = \lambda \int x_i(t)\ (y_d(t) - y_{\text{out}}(t))\ dt \tag{9}$$

Weights are updated in an iterative process. A training trail consists of a number of iterations or (epochs). In each epoch, all training samples are presented sequentially to the system. For each sample the Δw_i are computed and accumulated. After the presentation of all samples, the weights are updated to $w_i(e+1) = w_i(e) + \Delta w_i$, where e is the current epoch of the learning process.

Figure 1 presents an illustration of the working of the SPAN method. In this illustration, the neuron has three synapses connected to the input neurons. The weight of the synapses are initialized randomly. For the sake of simplicity, each input train consists of a single spike. However, the learning method can also deal with more than one spike per input neuron. The input pattern t_i^f is visualized in Figure 1a and its convolution with the α-kernel is in part b of the figure. The target (desired) pattern consists of two spikes $t_d^{(0)}$ and $t_d^{(1)}$.

Table 1. Tabular description of the neural parameters

LIF Neural Model	
Time constants	$\tau_m = 10$ms , $\tau_s = 5$ms
Membrane resistance	$R = 333.33\text{M}\Omega$
Spike threshold	$\vartheta = 20$mV
Reset potential	$u_r = 0$mV
Refractory period	$\tau_{\text{ref}} = 3$ms

Figure 1d,e depicts a graphical illustration of Equation 9. The presented stimulus causes the neuron to fire three output spikes at times $t_{\text{out}}^{(0)}$,$t_{\text{out}}^{(1)}$ and $t_{\text{out}}^{(2)}$, respectively. One of them $t_{\text{out}}^{(0)}$ equals the desired spike time $t_d^{(0)}$, Figure 1c. Thus, the wrong spikes will produce errors as shown in Figure 1d. Consequently, the error will be translated into a weight adjustment for w_0 and w_2, Figure 1e.

We define the area under the curve of the difference $y_d(t) - y_{\text{out}}(t)$ as an error between actual and desired output:

$$E = \int |y_d(t) - y_{\text{out}}(t)| \; dt \tag{10}$$

Clearly, the value of the error will be zero when $y_d(t)$ equals $y_{\text{out}}(t)$, this is when the corresponding spike trains are equal.

3 Experiments

Two main experiments conducted to demonstrate the main characteristics of SPAN. The first experiment is precise-time spike train generation. The second experiment is to measure the memory capacity of SPAN.

In both experiments, the network architecture consists of single neuron driven by n synapses. The parameters setting of the spiking neuron and synapses are summarized in Table 1. The input spike patterns stimulating the neuron are generated randomly. Each pattern has a number of spike trains equals the number of synapses. Each train consists of one spike generated randomly in the time interval (0, 200 ms). The simulation is performed using the NEST simulator [5].

3.1 Precise-Time Spike Pattern Association

The purpose of the first experiment is to demonstrate the concept of the proposed learning method. The task is to learn a mapping of a random input spike pattern to specific target output spike train. The desired spike train consists of five spikes at times $t_d = 33, 66, 99, 132, 165$ ms. Initially, the synaptic weights are generated uniformly in the range $(0, 10$ pA). The learning process is run for a maximum number of 100 epochs and the experiment is repeated for 100 trials with different random weight initialization and input pattern. The setup of the experiment is shown in Figure 2.

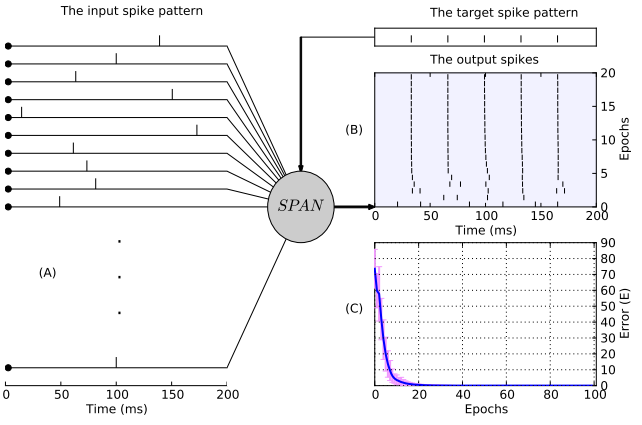


Fig. 2. Learning spike pattern association with 400 input synapses. The neuron learns to map between spatio-temporal input pattern and output spike train. (B) The development of the output toward the target pattern for one of the trials. (C) The evolving of the error and standard deviation.

SPAN is able to learn reproducing the desired output with high accuracy as shown in Figure 2b,c. We note that the neuron is able to reproduce the desired spike output pattern very precisely. In 97% of all trials the target spike train could be reproduced in less than 30 epochs and even for the remaining three percent the average temporal difference between learned and desired spike train was less than 0.2 ms.

3.2 Memory Capacity

Here we use a measure for the memory capacity of the spiking neuron proposed in [7]. The capacity is described as a so-called load factor α which is defined as the ratio of the number of input patterns p over the number of synapses n, *i.e.* $\alpha = \frac{p}{n}$. We note that according to this definition, increasing the number of synapses will allow the neuron to learn and recognize more patterns. The memory capacity is studied using different values for p and n. The p input patterns are generated randomly, similar to the previous experiments, and assigned randomly to $c = 5$ different classes. In a maximum number of 500 epochs, the neuron is trained to fire a single desired spike at a specified time $t_d^{(i)}$ which is associated with a class i. The specified times $t_d^{(i)}$ are set to either 33, 66, 99, 132, or 165 ms resulting in one spike time for each of the five classes. We consider a pattern as correctly classified, if the corresponding output spike is within 2 ms of the target spike. The learning rate was set to $\lambda = \frac{c}{p}$. The synaptic weights were initialized randomly following a uniform distribution and maximum values were scaled to 5, 2.5 and 2 pA for 200, 400 and 600 synapses respectively. This configuration is based on experimental observations and further investigations will be conducted in a future study in order to derive some practical guidelines to set these parameters properly.

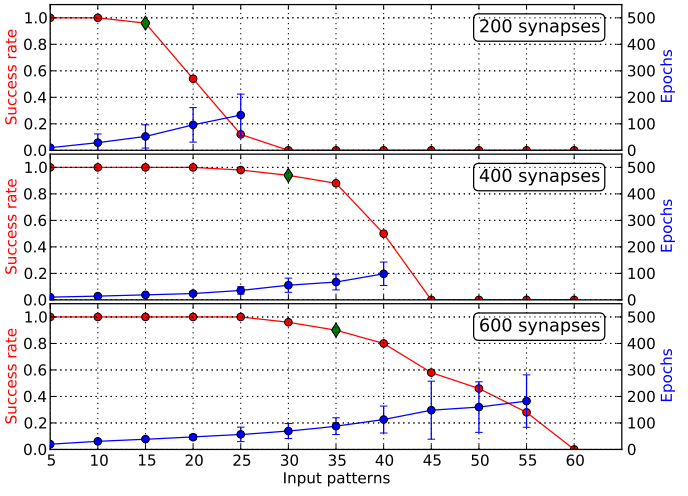


Fig. 3. The memory capacity of SPAN with different number of synapses

Figure 3 shows the results of the experiment averaged over 50 trials. We report the success rate which is defined as the number of trials in which all input patterns are classified correctly. For these successful trials, the number of epochs required to learn the correct classification result is shown as a separate curve in the diagrams. Clearly, increasing the number of synapses improves the memory capacity of the SPAN trained neuron. However, more number of epochs, and hence more computation time, is required. To get indication about the load factor, we consider the points where the success rate is 90% and above, indicated by the green diamond markers in Figure 3. The load factor at these points are 0.075, 0.075 and 0.058 for 200, 400 and 600 synapses respectively. We have compared these results to the ones reported in [3]. Our load factors are higher compared to the ReSuMe learning rule for which a load factor between 0.02 and 0.04 was obtained. Considering the small number of training epochs, our results are comparable to the Chronotron learning method.

4 Conclusion and Future Directions

While computation with spike is promising paradigm, developing learning methods is difficult. To overcome this difficulty we propose to transform the spike trains into analog signals to make it easier to compute the error during learning. SPAN, a LIF neuron proposed in this paper makes use of this idea. It converts the output and desired spike trains into continuous value signals and then the Delta rule is applied in a gradient decent mode. Further research will consider the application of SPAN for the design of more complex SNN and also for real world pattern recognition problems.

Acknowledgements. The work on this paper has been supported by the Knowledge Engineering and Discovery Research Institute (KEDRI, www.kedri.info). One of the authors, NK, has been supported by a Marie Curie International Incoming Fellowship with the FP7 European Framework Programme under the project EvoSpike, hosted by the Neuromorphic Cognitive Systems Group of the Institute for Neuroinformatics of the ETH and the University of Zurich.

References

1. Bohte, S.M.: The evidence for neural information processing with precise spike-times: A survey. Natural Computing 3 (2004)
2. Bohte, S.M., Kok, J.N., Poutré, J.A.L.: SpikeProp: backpropagation for networks of spiking neurons. In: ESANN, pp. 419–424 (2000)
3. Florian, R.V.: The chronotron: a neuron that learns to fire temporally-precise spike patterns (November 2010), http://precedings.nature.com/documents/5190/version/1, http://precedings.nature.com/documents/5190/version/1
4. Gerstner, W., Kistler, W.M.: Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge (2002)
5. Gewaltig, M.O., Diesmann, M.: Nest (neural simulation tool). Scholarpedia 2(4), 1430 (2007)
6. Goodman, E., Ventura, D.: Spatiotemporal pattern recognition via liquid state machines. In: International Joint Conference on Neural Networks, IJCNN 2006, Vancouver, BC, pp. 3848–3853 (2006)
7. Gutig, R., Sompolinsky, H.: The tempotron: a neuron that learns spike timing-based decisions. Nat. Neurosci. 9(3), 420–428 (2006), http://dx.doi.org/10.1038/nn1643
8. Legenstein, R., Naeger, C., Maass, W.: What can a neuron learn with spike-timing-dependent plasticity? Neural Computation 17(11), 2337–2382 (2005)
9. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation 14(11), 2531–2560 (2002)
10. Ponulak, F., Kasiński, A.: Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting. Neural Computation 22(2), 467–510 (2010) PMID: 19842989
11. van Rossum, M.C.: A novel spike distance. Neural Computation 13(4), 751–763 (2001)
12. Victor, J.D., Purpura, K.P.: Metric-space analysis of spike trains: theory, algorithms and application. Network: Computation in Neural Systems 8(2), 127–164 (1997), http://informahealthcare.com/doi/abs/10.1088/0954-898X_8_2_003

Induction of the Common-Sense Hierarchies in Lexical Data

Julian Szymański[1] and Włodzisław Duch[2,3]

[1] Department of Computer Systems Architecture, Gdańsk University of Technology, Poland
julian.szymanski@eti.pg.gda.pl
[2] Department of Informatics, Nicolaus Copernicus University, Toruń, Poland
[3] School of Computer Engineering, Nanyang Technological University, Singapore
Google: W. Duch

Abstract. Unsupervised organization of a set of lexical concepts that captures common-sense knowledge inducting meaningful partitioning of data is described. Projection of data on principal components allow for identification of clusters with wide margins, and the procedure is recursively repeated within each cluster. Application of this idea to a simple dataset describing animals created hierarchical partitioning with each clusters related to a set of features that have common-sense interpretation.

Keywords: hierarchical clustering, spectral analysis, PCA.

1 Introduction

Categorization of concepts into meaningful hierarchies lies at the foundation of understanding their meaning. Ontologies provide such hand-crafted hierarchical classification, but they are based usually on expert knowledge, not on the common-sense knowledge. For example, most biological taxonomies are hard to understand for lay people. There is no relationship between linguistic labels and their referents, so words may only point at the concept, inducing brain states that contain semantic information, predisposing people to meaningful associations and answers. In particular visual similarity is not related to names. Dog's breeds are categorized depending on their function, like Sheepdogs and Cattle Dogs, Scenthounds, Pointing Dogs, Retrievers, Companion and Toy Dogs, with many diverse breeds within each category. Such categories may have very little in common when all properties are considered. Differences between two similar dog breeds may be based on rare traits, not relevant to general classification. This makes identification of objects by their description quite difficult and the problem of forming common sense natural categories worth studying.

In this paper we have focused on relatively simple data describing animals. First, this is a domain where everyone has relatively good understanding of similarity and hierarchical description, second there is a lot of structured information in the Internet resources that may be used to create detailed description of the animals, third one can test the quality of such data by playing word games. We shall look at the novel way of using principal component analysis (PCA) to create hierarchical descriptions, but many other choices and other knowledge domains (for example, automatic classification of library subjects) may be treated using similar methodology.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 726–734, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 The Data

The data used in the experiments has been obtained using automatic knowledge acquisition followed by corrections resulting from the 20-questions word game [1]. The point of this game is to guess the concept the opponent is thinking of by asking questions that should narrow down the set of likely concepts. In our implementation[1] the program is trying to make a guess asking people questions. Results are used to correct lexical knowledge and in its final stage controlled dialog between human and computer, based on several plausible scenarios, is added to acquire additional knowledge. If the program wins, guessing the concept correctly, it will strengthen the knowledge related to this concept. If it fails, human is asked additional question „What did you think of?" and concepts related to the answer are added or features are modified according to the information harvested during the game.

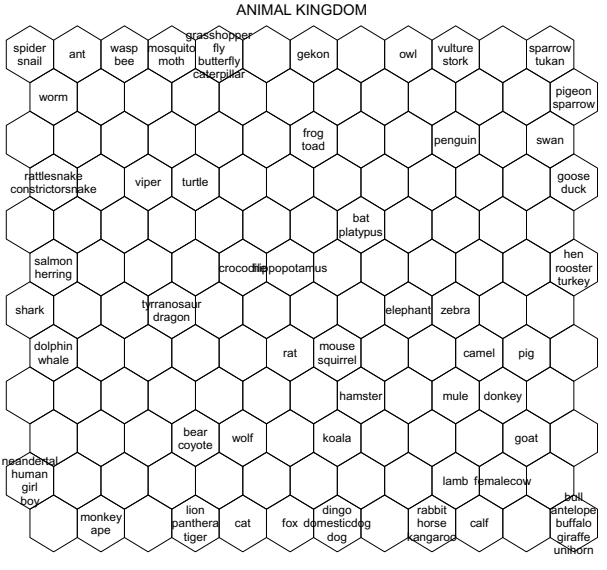


Fig. 1. Data used in the experiments visualized with Self-Organizing Map

Implementation of our knowledge acquisition system based on the 20-question game uses a semantic memory model [2] to store lexical concepts. This approach makes it more versatile than using just correlation matrix, as it has been successfully done in the implementation of this word game[2]. The matrix stores correlations between objects and features using weights that describe mutual association derived from thousands of games, providing decomposition of each concepts into a sum of contributions from questions. Such representation is flat and does not treat lexical features as natural

[1] http://diodor.eti.pg.gda.pl
[2] http://www.20-q.net

language concepts that allow for creation of a hierarchy of the common sense objects. Our program, based on semantic memory representation, shows elementary linguistic competence collecting common sense knowledge in restricted domains [1], and the knowledge generated may be used in many ways, for example by generating word puzzles.

This lexical data in semantic memory may be reorganized in a way that will introduce generalizations and increase cognitive economy [3]. This hierarchy is induced searching for the directions with highest variance using PCA eigenvectors, separating subsets of concepts and repeating the process to create consecutive subspaces. To illustrate and better understand this process a relatively small experiment has been performed.

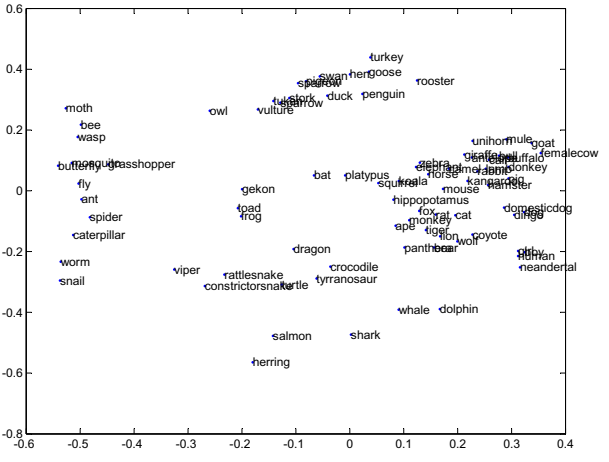


Fig. 2. The data used in the experiments visualized with MDS

A test dataset with 84 concepts (animals, or in general some objects) described by 71 features has been constructed after performing 346 games. The dataset used in the experiments is displayed using Self-Organizing Map (SOM) [4] visualization in Fig. 1 and with parametric Multidimensional Scaling (MDS) [5] in Fig. 2. Distances between points that represent dissimilarities between animals are calculated using cosine measures $d(\mathbf{X}, \mathbf{Z}) = \mathbf{X} \cdot \mathbf{Z} / ||\mathbf{X}|| ||\mathbf{Z}||$

3 PCA Directions

Expert taxonomies are frequently based on single feature, such as mammals, and then marsupials, but common-sense categorization is based on combination of features that makes objects similar. Principal Component Analysis [6] finds directions of highest data variance. Projecting the data on these direction shows interesting combination of features and thus helps to select groups of correlated features that separate data points, creating subsets of animals.

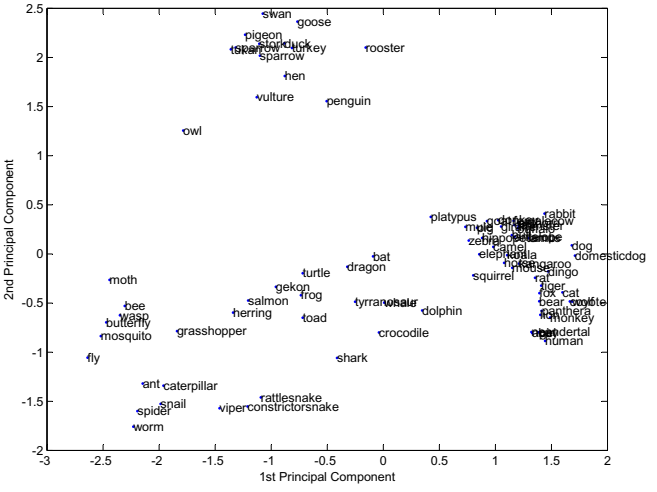


Fig. 3. Dataset visualization using two highest Principal Components

A pair of PCA directions may be used for visualization of the data. Projection on the first two directions with largest variance is shown in Fig. 3. The three visualizations (Figure 1, 2, 3) show different aspects of the data. Note for example the cluster formed in SOM containing *lion, pantera* and *tiger*. MDS shows their similarity but can still distinguish between them, while in PCA projection some objects appear between them (*fox, bear*) that do not fall into that cluster. PCA is able to find groups of related features and thus extract some commonsense knowledge approximating meaningful directions in the feature space. At the one end of the axis objects that have a mixture of features making them similar to each other are placed, on the other end objects that do not have such features. Fig. 4 shows coefficients of features in our semantic space for the first six principal components. Each feature, such as *lay-eggs, is-mammal* is placed above the line in one of the 6 columns, one for each component, to indicate the value of its coefficient in PCA vector. The most important features (having the highest absolute coefficient weights) in terms of data partitioning can be obtained from subsequent components. In the first vector (lowest row) the most negative (leftmost) coefficients correspond to features *lay-eggs, has-wings* describing insects and birds, while the most positive (right-most) are for *is-mammal, has-teeth, has-coat, is-warmblooded*, and others typical for mammals. The second PCA component has most positive coefficients for *has-beak, has-bill, has-feathers, is-bird, has-wings* indicating that this group of features is characteristic for the birds.

Hierarchical clusterization for such groups of features should show interesting common-sense clusters. In Fig. 5 direct projection of all vectors describing animals on each of these principal directions is shown. These projections show different aspects of the data, for example the projection on the second PCA shows a clear cluster for birds, starting with *swan* and ending with *owl* as less typical bird, the third cluster starts with *vulture* and groups other hunting animals. Projection on each PCA component may be used to generate different partitions of all objects.

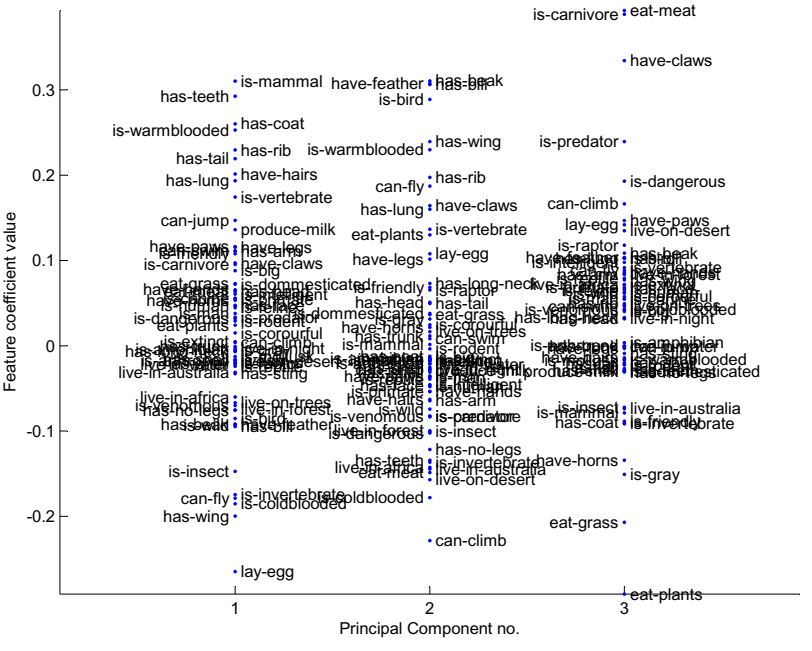


Fig. 4. Groups of features related to the principal components

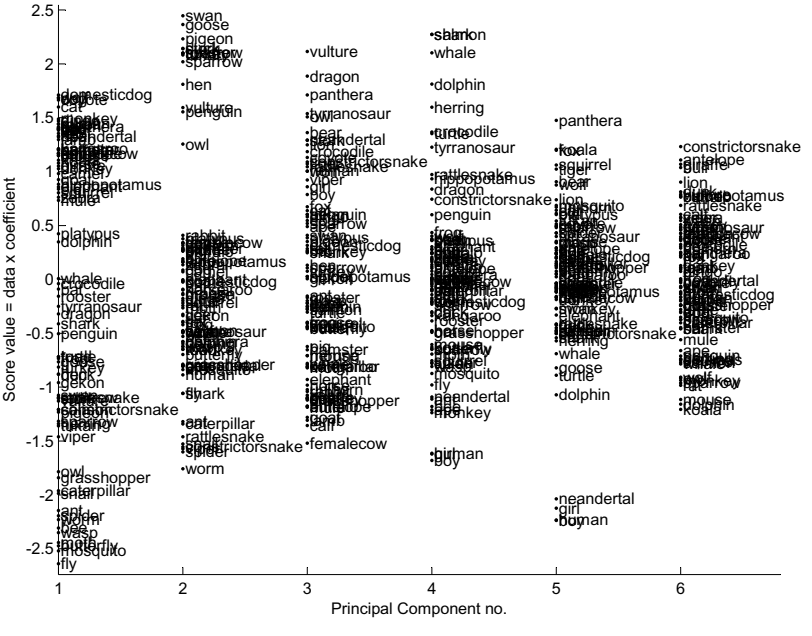


Fig. 5. Projections of the data on the first 6 principal components

4 Creating Hierarchical Partitioning

4.1 Hierarchical Agglomerative Partitioning

Creating a hierarchy based on similarity is one of the most effective ways for presenting large sets of concepts. Clustering data using agglomerative approach [7] is most frequently used for showing hierarchical organization of the data. The bottom-up approach using average linkage between clusters on each hierarchy level is shown in Fig. 6.

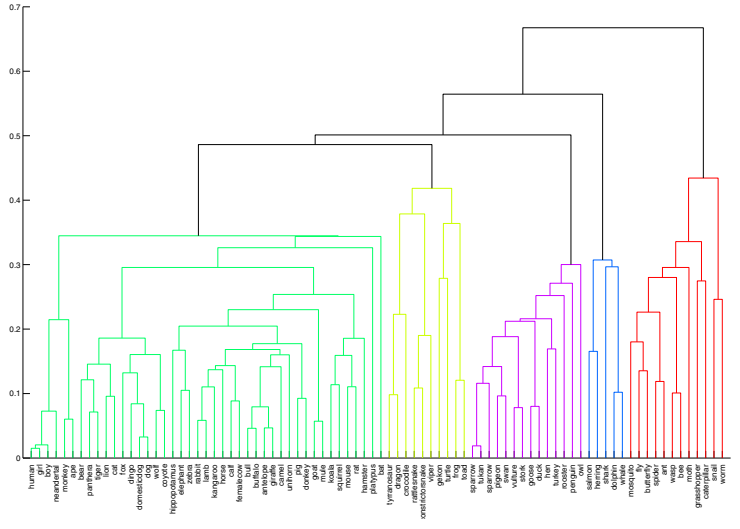


Fig. 6. Dendogram for animal kingdom dataset

Hierarchical agglomerative clustering using bottom-up approach binds together groups of objects in a way that frequently does not agree with intuitive partitioning. Moreover, the features used to construct a cluster are not easily traceable.

4.2 Hierarchical Partitioning with Principal Components

The distribution of the data points using the first 6 principal components (Fig. 5) shows a large gap between two groups projected with the second principal component. These two clusters of data are separated with the largest margin and thus should be meaningful. Hierarchical organization of the data can be analyzed from the point of view of graph theory. In terms of the graph bisection the second eigenvector is most important [8], allowing for creation of normalized cut (partition of the vertices of a graph into two disjoint subsets) [9]. Thus selecting the second principal component is a good start to construct hierarchical partitioning. The typical approaches to spectral clustering employs second (biggest) component [10] (that minimize graph conductance) or a second smallest component [11] (due to Rayleigh theorem).

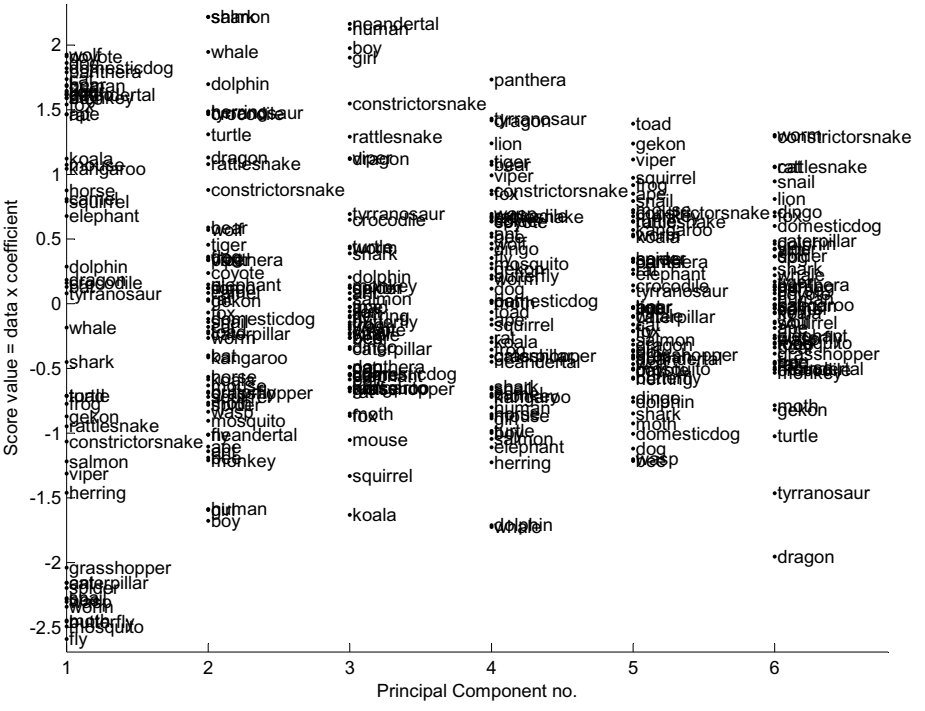


Fig. 7. Projections of the reduced data set using succeeding single principal components

Analyzing subsequent PCA component projections (given in Fig. 5 and 7) shows that the second principal component does not lead always to the best cut in the graph. It is better to select the component that produces the widest separation margin within the data, choosing a different principal component for each hierarchy level. For creating the first hierarchy level the second component is selected, separating birds from other animals, creating one pure and one mixed cluster (Fig. 5). Features of the second PCA component (Fig. 4) with lowest and highest weights include: (-)*climb*, (-)*cold-blooded* and (+)*beak*, (+)*feather*, (+)*bird*, (+)*wing*, (+)*warmblooded*. Note that one feature *is-bird* alone is sufficient to create this partitioning but correlated features separate this cluster in a better way.

To capture some common-sense knowledge hierarchical partitioning is created in a top-down way Each of the newly created clusters is analyzed using PCA and principal components that give the widest separation margins are selected for data partitioning. PCA is performed recursively on reduced data that belong to the selected cluster. In Fig. 7 the first 6 components computed for the large mixed cluster (that does not contain birds) created on the second hierarchy level is presented. This cluster has been formed after separating the birds and other animals with the second component (shown in Figure 5). Within this cluster the widest margin is created with the first component and it separate mammals (with the exception of dolphins and whales) from other animals.

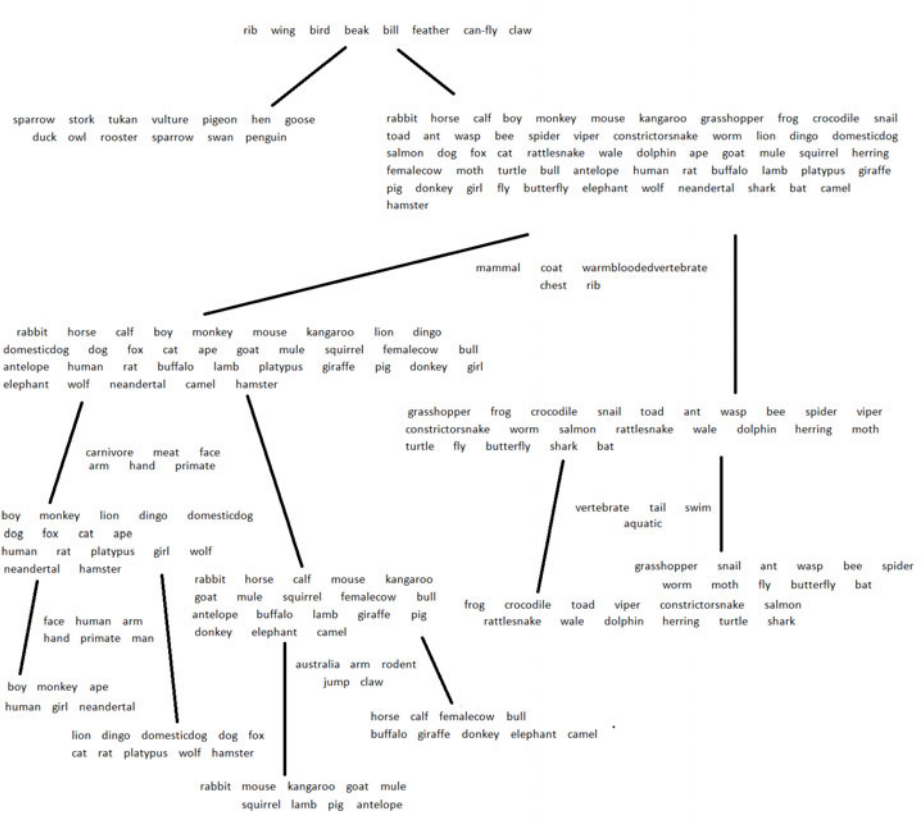


Fig. 8. Hierarchy of the data and features used to create it

Repeating the process described above hierarchical organization of the data is introduced. In Fig. 8 a top part of created hierarchy is shown. At each level of the hierarchy most important features used to create this partition are also displayed.

5 Discussion and Future Directions

An approach to create hierarchical commonsense partitioning of data using recursive Principal Component Analysis has been presented. Results of this procedure have been illustrated on simple data describing animals created using the 20-questions game that is based on model of semantic memory [1]. This approach has been used for creating general clusters within the semantic memory model that stores natural language concepts. Such analysis allows for finding additional correlations between features facilitating associative processes for existing concepts, and improving the learning proces when new information is added to the system. In the neurolinguistic approach to the natural language processing [12] it has been conjectured that the right brain hemisphere creates receptive fields (called “cosets", or constraint-sets) that constrain semantic interpretation, although they do not have linguistic labels themselves. The process describe

here may be an approximation ot some of the neural processes responsible for language comprehension.

Hierarchical organization of lexical data has been created here in an unsupervised way by selecting linear combinations of features that provide clear separation of concepts. Extension of this approach may be based on bi-clustering, taking into account clusters of features that are relevant for creating meaningful clusters of data. The main idea is to strengthen features that are correlated to the dominant one, or to the features given by the user who may want to view the data from a specific angle [13]. Non-negative matrix factorization [14] is another useful technique that may replace PCA. Many other variants of unsupervised data analysis methods are worth exploring in the context of this approach to induction of the common-sense hierarchies in data.

Acknowledgements. The work has been supported by the Polish Ministry of Science and Higher Education under research grant N519 432 338.

References

1. Szymański, J., Duch, W.: Information retrieval with semantic memory model. Cognitive Systems Research (in print, 2011)
2. Tulving, E.: Episodic and semantic memory. Organization of Memory, 381–402 (1972)
3. Conrad, C.: Cognitive economy in semantic memory (1972)
4. Kohonen, T.: The self-organizing map. Proceedings of the IEEE 78, 1464–1480 (1990)
5. Shepard, R.: Multidimensional scaling, tree-fitting, and clustering. Science 210, 390 (1980)
6. Jolliffe, I.: Principal component analysis. Wiley Online Library (2002)
7. Day, W., Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods. Journal of Classification 1, 7–24 (1984)
8. Rahimi, A., Recht, B.: Clustering with normalized cuts is clustering with a hyperplane. Statistical Learning in Computer Vision (2004)
9. Dhillon, I.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 269–274. ACM (2001)
10. Kannan, R., Vetta, A.: On clusterings: Good, bad and spectral. Journal of the ACM (JACM) 51, 497–515 (2004)
11. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)
12. Duch, W., Matykiewicz, P., Pestian, J.: Neurolinguistic approach to natural language processing with applications to medical text analysis. Neural Networks 21(10), 1500–1510 (2008)
13. Szymański, J., Duch, W.: Dynamic Semantic Visual Information Management. In: Proceedings of the 9th International Conference on Information and Management Sciences, pp. 107–117 (2010)
14. Lee, D., Seung, S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)

A Novel Synthetic Minority Oversampling Technique for Imbalanced Data Set Learning

Sukarna Barua[1], Md. Monirul Islam[1], and Kazuyuki Murase[2]

[1] Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh
[2] University of Fukui, Fukui, Japan

Abstract. Imbalanced data sets contain an unequal distribution of data samples among the classes and pose a challenge to the learning algorithms as it becomes hard to learn the minority class concepts. Synthetic oversampling techniques address this problem by creating synthetic minority samples to balance the data set. However, most of these techniques may create wrong synthetic minority samples which fall inside majority regions. In this respect, this paper presents a novel Cluster Based Synthetic Oversampling (CBSO) algorithm. CBSO adopts its basic idea from existing synthetic oversampling techniques and incorporates unsupervised clustering in its synthetic data generation mechanism. CBSO ensures that synthetic samples created via this method always lie inside minority regions and thus, avoids any wrong synthetic sample creation. Simualtion analyses on some real world datasets show the effectiveness of CBSO showing improvements in various assesment metrics such as overall accuracy, F-measure, and G-mean.

Keywords: Imbalanced learning, Unsupervised clustering, Synthetic oversampling.

1 Introduction

An imbalanced data set has unequal distribution of samples among the classes. The class having the majority of the data samples is called the majority class and the other, minority class. Classifiers usually aim to reduce the global classification error and therefore, any classifier, learned from an imbalanced dataset, shows greater classification errors over the examples of minority class [1,2]. This becomes very costly in many real world problems such as information retrieval [3], detection of fraudulent telephone calls [4] and oil spills in radar images [5], data mining from direct marketing [6], and helicopter fault monitoring [7], where the identification of the minority is of utmost importance.

Significant works have been conducted to deal with the imbalanced learning problem. Most of these works fall into one of the four different categories: sampling based approaches, cost based approaches, kernel based approaches, and active learning based approaches [8]. In this paper, we are only interested in sampling based approaches and therefore, we provide a brief overview of the works performed in this category only. Details of works performed in other categories can be found in [8].

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 735–744, 2011.
© Springer-Verlag Berlin Heidelberg 2011

In imbalanced learning, sampling methods modify an imbalanced data set to create a balanced data set. Balanced data sets perform well than imbalanced data sets for many base classifiers [9,10]. There are two different types of sampling methods: undersampling and oversampling. Undersampling methods work by reducing the number of instances of the majority class either randomly or by using some statistical knowledge to balance the class distribution [11,12,13,14]. On the other hand, oversampling methods add minority samples by random resampling of the original minority class [15,16] or by creating synthetic samples for the minority class. Depending on the technique of how synthetic samples will be generated, there are many variants existing in literature such as Synthetic Minority Oversampling Technique (SMOTE) [17], Borderline-SMOTE [18] and Adaptive Synthetic Sampling Technique (ADASYN) [19]. Application of boosting [20] can be integrated with sampling to provide a better classifier structure for imbalanced learning. Works in this category were proposed by SMOTEBoost [21], DataBoost-IM [22], RAMOBoost [23], etc.

Although both of undersampling and oversampling approaches have been shown to improve classifer performance over imbalanced data sets, in [24], it was shown that, oversampling is lot more useful than undersampling. The performance of oversampling algorithms was shown to improve dramatically even for complex data sets [24].

In this paper, we propose a new technique Cluster Based Synthetic Oversampling (CBSO) algorithm. CBSO integrates clustering with the data generation mechanism of existing synthetic oversampling techniques. While most of the existing techniques may create wrong synthetic minority samples falling inside majority regions, CBSO avoids it by ensuring that synthetic samples created by CBSO always lie inside minority regions. In this way, CBSO creates a best set of synthetic samples to balance the class distribution.

The remainder of this paper is divided into four sections. In Sect. 2, we present the motivation behind CBSO. Section 3 describes the details of CBSO algorithm. In Sect. 4, we present the experimental study and simulation results. Finally, in Sect. 5, we provide some future aspects of this research and conclude the paper.

2 Motivation

Synthetic oversampling techniques such as SMOTE [17], Borderline-SMOTE [18], and ADASYN [19] have been shown to be very successful in dealing with imbalance data sets. However, our study finds out some insufficiencies and inappropriatenesses of these existing techniques that may occur in many different scenarios of the data samples which are described below.

In data generation phase, most of the existing synthetic oversampling techniques employ a k-nearest neighbor based (k-NN) approach. In this approach, to create a new synthetic sample from an existing minority sample x, another minority sample y is randomly selected from the k nearest neighbors of x (where

k is a user specified parameter), and a synthetic sample g is generated, by linear interpolation of x and y:

$$g = x + (y - x) \times \alpha \tag{1}$$

where α is a random number in the range $[0, 1]$. Equation (1) says that, the generated synthetic sample g lies in the line segment between x and y. However, in many scenarios, this k nearest neighbor based data generation approach may lead to creation of wrong minority samples. To show why, consider Fig. 1, where stars and circles represent majority and minority samples, respectively. For this figure, assume that, we are creating a synthetic sample from minority sample, A. Assuming $k = 5$, k-NN approach will randomly select another minority sample from the 5 nearest minority neighbors of A, say B is selected. Now, linear interpolation (1) of A and B may result in the generation of a synthetic sample like P shown by a square in Fig. 1. We see that, created sample P is clearly a wrong minority sample, because it overlaps with a majority sample in the figure.

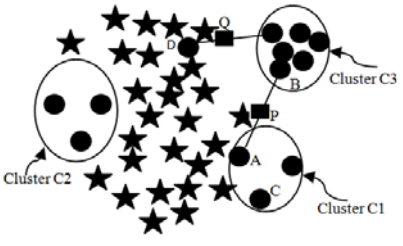


Fig. 1. Figure illustrating problems of k-nearest neighbor based data generation approach

The above problem is magnified when small sized clusters are present in minority class concept. Consider samples of cluster $C1$ and cluster $C2$ in Fig. 1. If synthetic samples are created from any member x of any of these clusters (say cluster $C2$), it is likely that, k-NN approach will select y of (1) from the other cluster (cluster $C1$), resulting in the generation of synthetic samples in the majority region between the two minority clusters. The generated samples will clearly create overlapping minority and majority regions which will make the learning task harder. The above problem is even worse when synthetic samples are generated from a noisy sample such as sample D in Fig. 1 (Q might be a synthetic sample created which falls inside majority regions in the figure).

So, from the above discussion, we conclude that, k-NN data generation approach may create wrong minority samples. The problem occurs due to the fact that, the approach uses all k nearest neighbors blindly without considering their position and distance from the minority sample under consideration. Morever, the appropriate value of k cannot be determined, beforehand.

3 Proposed CBSO Algorithm

Motivated by problems stated in Sect. 2, we have devised a new minority oversampling technique which we call Cluster Based Synthetic Oversampling (CBSO) algorithm. CBSO combines the synthetic oversampling mechanism of existing techniques with a different data generation mechanism based on clustering. The complete CBSO algorithm is shown in [Algorithm CBSO]. The basic approach of CBSO (Steps 1 to 4 of [Algorithm CBSO]) is adopted from the state of the art ADASYN [19] algorithm. CBSO differs from ADASYN in Steps 5 and 6 where synthetic data samples are generated using an unsupervised clustering technique rather than k-NN approach (as used in ADASYN). The details of synthetic data generation mechanism of CBSO (Steps 5 and 6) are discussed below.

[**Algorithm CBSO**]

Input:
Training data samples D_{tr} with m samples $\{x_i, y_i\}$, $i = 1 \cdots m$, where x_i is an instance in n dimensional feature space X, and $y_i \in \{-1, 1\}$ is the class identity level associated with x_i. Deifne m_s and m_l as the number of minority class examples and number of majority class examples, respectively. Therefore, $m_s \leq m_l$ and $m_s + m_l = m$.

Procedure:

1. Calculate the number of synthetic samples that need to be generated for the minority class:
$$G = (m_l - m_s) \times \beta$$
Where $\beta \in [0, 1]$ is a parameter used to specify the desired balance level after generation of the synthetic data. $\beta = 1$ means a fully balanced dataset is created after the generation process.
2. For each sample x_i in *minorityclass*, find K nearest neighbors based on the Euclidean distance in n dimensional space and calculate the ratio r_i defined as:
$$r_i = \Delta_i / K, i = 1 \cdots m_s$$
Where Δ_i is the number of samples in K nearest neighbors of x_i that belong the majority class, therefore, $r_i \in [0, 1]$.
3. Normalize r_i according to $\widehat{r_i} = r_i / \sum_{i=1}^{m_s} r_i$, so that $\widehat{r_i}$ is a density distribution ($\sum \widehat{r_i} = 1$)
4. Calculate the number of synthetic samples g_i that need to be generated for each minority sample x_i:
$$g_i = \widehat{r_i} \times G$$
5. Find the clusters of *minorityclass*
6. For each minority sample x_i generate g_i synthetic data samples according to the following steps:
Do the **loop** from $1 \cdots g_i$

(a) Randomly select one minority sample y, from x_i's cluster (as found in Step 5).
(b) Generate the synthetic data, s according to $s = x_i + \alpha \times (y - x_i)$, where α is a random number in the range $[0, 1]$.

End **Loop**

3.1 Synthetic Data Generation Mechanism of CBSO

In Sect. 2, we showed the problems of k-nearest neighbor based data generation. To improve this, CBSO adopts a different data generation mechanism based on unsupervised clustering (Steps 5 and 6 of [Algorithm CBSO]). Step 5 finds the clusters of the *minorityclass*. In Step 6(b), CBSO changed the way, how y of (1) is chosen for x_i. Rather than, choosing y as one, at random, from the k nearest neighbors of x_i (as in k-NN approach), CBSO selects y as one from x_i's cluster (as found in Step 5 of [Algorithm CBSO]). The intuition is that, if y is selected as one from the cluster of x_i, then synthetic samples that will be generated from x_i and y according to (1) will also lie inside the same cluster whose member is x_i. Therefore, created samples will never fall in majority regions. Another advantage of CBSO is that, noisy minority samples will likely form an isolated cluster during the clustering process in Step 5 ([Algorithm CBSO]). Hence, if synthetic samples are created from such a noisy sample x, y of (1) will be the same sample x (since, x is the only member of its cluster). So, created samples will be a duplication of x, according to (1). This is much better than k-NN approach, which may create more noisy and wrong minority samples and broaden minority region. The broadened minority region by the k-NN approach may overlap with majority regions and may over generalize the classifier erroneously.

3.2 Clustering Minority Class

The success of CBSO will largely depend on how we cluster the *minorityclass* in Step 5 of [Algorithm CBSO]. For this purpose, CBSO uses average-linkage agglomerative clustering, a hierarchical clustering algorithm [25,26]. Agglomerative clustering does not require the number of clusters beforehand. The algorithm generates clusters in a bottom-up fashion. The key steps of this algorithm are given below (assume, N data samples are given as input):

1. Initially, each data sample is assigned to a seperate cluster. So, initially there are N clusters, each of size one.
2. Find the two closest clusters, cluster I and cluster J.
3. Merge cluster I and cluster J into a single cluster, cluster K. This merging reduces number of clusters by one.
4. Update distance measures between the newly computed cluster and all previous clusters.
5. Repeat steps 2-4 until all data samples are merged into a single cluster of size N.

The basic algorithm described above produces one cluster of size N, which is, definitely, not our goal. We can find more than one cluster, if we stop the merging process in Step 3 early. For this purpose, CBSO uses a threshold, T_h and stops the merging process when the distance between closest pair exceeds this threshold. The output will be the set of clusters remaining at that point of the algorithm.

What should be value of T_h? Clearly, this value should not be constant, because, the distance measure varies with dimension of the feature space. So, the same algorithm will produce different number of clusters for the same types of data sets, where the only difference is in feature space dimension. The second problem of using a constant value for T_h lies in the fact that in some data sets samples are relatively sparse (average distance between samples is high), whereas in some other data sets, samples are relatively dense (average distance between samples is low). So, using a constant T_h, will produce, fewer number of clusters for data sets where average distance between samples is low and larger number of clusters for data sets where average distance between samples is high. So, the intuition is that, the value of T_h should be data set dependent. It should be calculated using some heuristics of the distance measures between samples of the data set. For this purpose, we first find a value d_{avg} as follows:

$$d_{avg} = \frac{1}{|minorityclass|} \sum_{x \in minorityclass} \min_{y \neq x, y \in minorityclass} \{dist(x, y)\}$$

We then compute T_h by multiplying d_{avg} by a constant parameter, C_p:

$$T_h = d_{avg} \times C_p$$

For each member of *minorityclass*, we find the minimum Euclidean distance to any other member in the same class. We, then find the average of all these minimum distances, to form, d_{avg}. Parameter C_p is used to tune the output of clustering algorithm. Increasing value of C_p, increases cluster sizes, reducing the number of clusters. Decreasing the value of C_p, decreases cluster size, increasing number of clusters generated.

4 Experimental Study

In this section, we evaluate the effectiveness of our proposed CBSO algorithm and compare its performance with SMOTE [17] and ADASYN [19] algorithm. We evaluate the performance of these three oversampling techniques using two different base classifier models: backpropagation neural network and decision tree classifier [27]. We use eight datasets from UCI machine learning repositorty [28]. Some of these original data sets were multi-class data. Since, we are only interested in two-class classification problem, these data sets were transformed to form two-class data sets to ensure a minimum level of imbalance. Table 1 shows

Table 1. Description of data set characteristics used in simulation experiments

Dataset	Minority Class	Features	Instances	Minority	Majority	%Minority
Abalone	Class of '18'	7	731	42	689	6%
Vehicle	Class of '1'	18	940	219	721	23%
Glass	Class of '5','6','7'	9	214	51	163	24%
Wine	Class of '3'	13	178	48	130	27%
Texture	Class of '2','3','4'	40	5477	1500	3977	28%
Pima	Class of '1'	8	768	268	500	35%
Ionosphere	Class of 'Bad'	34	351	126	225	36%
Spambase	Class of 'Spam'	57	4601	1813	2788	40%

the minority class composition (these classes in the original dataset were combined to form the minority class and rest of the classes form the majority class) and other charactersitics of the data sets such as number of features, number of total instances in the dataset, and number of majority and minority instances.

As evaluation metrics, we use overall accuracy as one of our performance metrics as it is very well known and popular in research community [8]. However, accuracy depends on the distribution of positive and negative examples in the data set and may not reflect the actual performance in imbalanced data sets [8]. Therefore, besides accuracy, we use two other performance measures suitable for imbalanced learning: F-measure, and G-mean [8].

In the simulation experiments, we run single decision tree classifier and single neural network classifier on the eight datasets shown in Table 1. For the neural network classifier, the number of hidden neurons is randomly set to 5, number of input neurons is set to be equal to the number of features in the dataset and number of output neurons is set to 2. Sigmoid function is used as the activation function. Number of training epochs is randomly set to 300 and learning rate is set to 0.1. For both of SMOTE and ADASYN, the value of nearest neighbors, K is set to 5 [17,19]. For ADASYN, we set $\beta = 1$, and $d_{th} = 0.75$ [19]. For SMOTE, we set $N = 200$ [17]. For CBSO, the parameter settings are: $K = 5$, $\beta = 1$ and $C_p = 3$. We provide the simulation results of overall accuracy, F-measure, and G-mean in Table 2 where each result is found after a 10-fold cross-validation. Best results are highlighted in bold-face type.

For both classifier models, from Table 2, we find that, CBSO provides the best results in terms of accuracy, F-measure, and G-mean in most of the datasets. The total winning times for each algorithm are also shown in Table 2, where CBSO outforms SMOTE and ADASYN in all performance metrics. As described in Sect. 2, k nearest neighbor based data generation technique creates wrong synthetic minority samples, which enlarges the minority region erroneously, falling inside the majority region. Due to this overgeneralization of minority region, SMOTE and ADASYN misclassify many majority samples as minority, decreasing the value of accuracy, F-measure, and G-mean performance metrics.

Table 2. Performance of SMOTE [17], ADASYN [19], and CBSO on eight real world datasets of Table 1 using single Neural Network and single Decision Tree classifier

		Neural Network			Decision Tree		
Dataset	Method	Accuracy	F-measure	G-mean	Accuracy	F-measure	G-mean
Abalone	SMOTE	0.81789	0.36799	0.79808	0.91244	0.30641	0.49226
	ADASYN	0.80292	0.33241	0.78847	0.90151	0.2433	0.46667
	CBSO	**0.84537**	**0.39178**	**0.81468**	**0.91664**	**0.36592**	**0.5773**
Glass	SMOTE	0.92033	0.84147	0.90422	0.89735	0.78302	**0.84832**
	ADASYN	**0.93438**	0.88147	**0.94329**	0.87792	0.74485	0.83547
	CBSO	0.95301	**0.90101**	0.93323	**0.91124**	**0.79288**	0.8472
Ionosphere	SMOTE	0.8916	0.83352	0.85496	0.84634	0.79072	0.83345
	ADASYN	0.89969	0.8499	0.87301	0.8465	0.79923	0.84334
	CBSO	**0.90295**	**0.85746**	**0.88063**	**0.85753**	**0.80883**	**0.85231**
Pima	SMOTE	**0.73168**	0.66315	**0.7358**	0.67329	0.55784	0.64896
	ADASYN	0.71994	0.65494	0.72702	**0.68611**	0.56263	0.65391
	CBSO	0.72126	**0.66594**	0.73484	0.68242	**0.58869**	**0.67076**
Vehicle	SMOTE	0.94578	0.89342	0.94979	0.93823	0.86886	0.91185
	ADASYN	**0.95641**	**0.91514**	0.96167	**0.94674**	**0.88657**	**0.92725**
	CBSO	**0.95641**	0.91413	**0.96491**	0.94358	0.87874	0.91903
Wine	SMOTE	0.97712	0.95758	**0.97767**	0.96667	0.94313	0.96344
	ADASYN	0.97712	0.95758	**0.97767**	0.97745	0.96111	0.97751
	CBSO	**0.98301**	**0.96667**	0.97496	**0.99412**	**0.98889**	**0.99608**
Texture	SMOTE	0.90195	0.84362	0.91863	0.96677	0.93983	0.96018
	ADASYN	0.89081	0.83585	0.91591	**0.97626**	0.95672	0.97067
	CBSO	**0.91768**	**0.87014**	**0.92926**	0.97608	**0.95698**	**0.97437**
Spambase	SMOTE	0.91415	0.8911	0.91003	**0.88568**	**0.85765**	**0.88358**
	ADASYN	0.89633	0.8734	0.89821	0.87829	0.84978	0.87747
	CBSO	**0.91524**	**0.8949**	**0.91536**	0.87981	0.85074	0.87792
Winning Times	SMOTE	1	0	2	1	1	2
	ADASYN	1	1	2	3	1	1
	CBSO	**7**	**7**	**5**	**4**	**6**	**5**

5 Conclusion

In this paper, we present a new synthetic oversampling algorithm, which is called CBSO, for balancing majority and minority class distribution in an imblanced data set. CBSO adaptively finds the numeber of synthetic samples to be generated from each minority sample using a weight distribution, which is similar to the ADASYN [19] algorithm. However, unlike ADASYN, CBSO generates synthetic samples using an unsupervised clustering which ensures that the generated synthetic samples reside inside some minority region, thus, avoiding any wrong minority sample creation. The simulation results show that, CBSO can outperform existing algorithms in terms of a good number of overall performance measures such as accuracy, F-measure, and G-mean. Several other research issues can be investigated using CBSO such as applicaton of CBSO in multiclass problems, integration of CBSO with some other undersampling methods,

performance of different clustering approaches in CBSO, and integration of CBSO with an ensemble technique such as Adaboost.M2 boosting ensemble.

Acknowledgments. The work has been done as part of M.Sc. Engg. Thesis of Computer Science & Engineering Department of Bangladesh University of Engineering and Technology (BUET). The authors would like to acknowledge BUET for its generous support.

References

1. Weiss, G.M.: Mining with Rarity: A Unifying Framework. ACM SIGKDD Explorations Newsletter 6(1), 7–19 (2004)
2. Holte, R.C., Acker, L., Porter, B.W.: Concept Learning and the Problem of Small Disjuncts. In: Proc. Int'l J. Conf. Artificial Intelligence, pp. 813–818 (1989)
3. Lewis, D., Catlett, J.: Heterogenous Uncertainty Sampling for Supervised Learning. In: Proceedings of the Eleventh International Conference on Machine Learning, pp. 148–156 (1994)
4. Fawcett, T.E., Provost, F.: Adaptive Fraud Detection. Data Mining and Knowledge Discovery 3(1), 291–316 (1997)
5. Kubat, M., Holte, R.C., Matwin, S.: Machine Learning for the Detection of Oil Spills in Satellite Radar Images. Machine Learning 30(2/3), 195–215 (1998)
6. Ling, C.X., Li, C.: Data Mining for Direct Marketing: Problems and Solutions. In: International Conference on Knowledge Discovery & Data Mining (1998)
7. Japkowicz, N., Myers, C., Gluck, M.: A Novelty Detection Approach to Classification. In: Proceedings of the Fourteenth Joint Conference on Artificial Intelligence, pp. 518–523 (1995)
8. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21(10), 1263–1284 (2009)
9. Weiss, G. M., Provost, F.: The Effect of Class Distribution on Classifier Learning: An Empirical Study. Technical Report ML- TR-43, Dept. of Computer Science, Rutgers Univ. (2001)
10. Laurikkala, J.: Improving Identification of Difficult Small Classes by Balancing Class Distribution. In: Proc. Conf. AI in Medicine in Europe: Artificial Intelligence Medicine, pp. 63–66 (2001)
11. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory Under Sampling for Class Imbalance Learning. In: Proc. Int'l. Conf. Data Mining, pp. 965–969 (2006)
12. Zhang, J., Mani, I.: KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. In: Proc. Int'l. Conf. Machine Learning (ICML 2003), Workshop Learning from Imbalanced Data Sets (2003)
13. Kubat, M., Matwin, S.: Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. In: Proc. Int'l. Conf. Machine Learning, pp. 179–186 (1997)
14. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. ACM SIGKDD Explorations Newsletter 6(1), 20–29 (2004)
15. Mease, D., Wyner, A.J., Buja, A.: Boosted Classification Trees and Class Probability/Quantile Estimation. J. Machine Learning Research 8, 409–439 (2007)
16. Jo, T., Japkowicz, N.: Class Imbalances versus Small Disjuncts. ACM SIGKDD Explorations Newsletter 6(1), 40–49 (2004)

17. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-Sampling Technique. J. Artificial Intelligence Research 16, 321–357 (2002)
18. Han, H., Wang, W.Y., Mao, B.H.: Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Proc. Int'l. Conf. Intelligent Computing, pp. 878–887 (2005)
19. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In: Proc. Int'l. J. Conf. Neural Networks, pp. 1322–1328 (2008)
20. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: Proc. Int'l Conf. Machine Learning, pp. 148–156 (1996)
21. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: Improving Prediction of the Minority Class in Boosting. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 107–119. Springer, Heidelberg (2003)
22. Guo, H., Viktor, H.L.: Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost IM Approach. ACM SIGKDD Explorations Newsletter 6(1), 30–39 (2004)
23. Chen, S., He, H., Garcia, E.A.: RAMOBoost: Ranked Minority Oversampling in Boosting. IEEE Trans. Neural Networks 21(20), 624–1642 (2010)
24. Japkowicz, N., Stephen, S.: The Class Imbalance Problem: A Systematic Study. Intelligent Data Analysis 6(5), 429–449 (2000)
25. Voorhees, E.M.: Implementing Agglomerative Hierarchic Clustering Algorithms for use in Document Retrieval. Information Processing and Management 22(6), 465–476 (1986)
26. Schutze, H., Silverstein, C.: Projections for Efficient Document Clustering. In: SIGIR 1997: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, PA, USA (1997)
27. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann, San Francisco (1993)
28. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/

A New Simultaneous Two-Levels Coclustering Algorithm for Behavioural Data-Mining

Guénaël Cabanes[1], Younès Bennani[1], and Dominique Fresneau[2]

[1] CNRS, UMR 7030, Université Paris 13, LIPN
[2] LEEC, EA 4443, Université Paris 13
99 av. J.-B. Clément - 93430 Villetaneuse, France

Abstract. Clustering is a very powerful tool for automatic detection of relevant sub-groups in unlabeled data sets. It can be sometime very interesting to be able to regroup and visualize the attributes used to describe the data, in addition to the clustering of these data. In this paper, we propose a coclustering algorithm based on the learning of a Self Organizing Map. The new algorithm will thus be able at the same time to map data and features in a low dimensional sub-space, allowing simple visualization, and to produce a clustering of both data and features. The resulting output is therefore very informative and easy to analyze.

Keywords: Coclustering, Biclustering, Two Mode Clustering, Self-Organizing Map, Ants Behaviour.

1 Introduction

Unsupervised classification, or clustering, is a very powerful tool for automatic detection of relevant sub-groups or clusters in unlabeled data sets, when one does not have prior knowledge about the hidden structure of these data. Patterns in the same cluster should be similar to each other, while patterns in different clusters should not (internal homogeneity and the external separation). Clustering plays an indispensable role for understanding various phenomena described by data sets. A clustering problem can be defined as the task of partitioning a set of objects into a collection of mutually disjoint subsets. Clustering is a segmentation problem which is considered as one of the most challenging problems in unsupervised learning. Various approaches have been proposed to solve the problem [1].

However, it can be sometime very interesting to be able to regroup and visualize the attributes used to describe the data, in addition to the clustering of these data. This allows, for example, to combine in a simple way each cluster of data with the characteristic features of this cluster, but also to visualize correlations between attributes. Coclustering, biclustering, or two mode clustering is a data mining technique which allows simultaneous clustering of rows and columns of data sets (data matrix) [2]. Given a set of m rows in n columns (i.e., an $m \times n$ matrix), the coclustering algorithm generates coclusters - a subset of rows which exhibit similar behavior across a subset of columns, or vice versa. The most

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 745–752, 2011.
© Springer-Verlag Berlin Heidelberg 2011

popular application for such methods is gene expression analysis, i.e. to identify local patterns in gene expression data (see [3]).

In this paper, we propose a coclustering algorithm based on the learning of a Self Organizing Map (SOM) [4]. The SOM is a neuro-computational algorithm to map high-dimensional data to a two-dimensional space through a competitive and unsupervised learning process. This unsupervised learning algorithm is a popular nonlinear technique for dimensionality reduction and data visualization, with a very low computational cost. The use of SOM to perform coclustering have been proposed in [5,6]. However, in these works, each cluster is represented by an unique prototype of the SOM, which leads to an inappropriate number of clusters. The proposed method will combine a modified SOM with a two-level coclustering of the SOM prototypes ables to detect automatically the correct number of clusters. The new algorithm will thus be able at the same time to map data and features in a same low dimensional sub-space, allowing simple visualization, and to produce a clustering of both data and features. The resulting output is therefore very informative and easy to analyze.

The remainder of this paper is organized as follows. Section 2 presents the SOM algorithm. Section 3 describes the new SOM-based coclustering algorithm. In section 4 we show the result of an experimental study in biology obtained with our algorithm. Conclusions are given in section 5.

2 Self-Organizing Map Adaptation for Disjunctive Data

Kohonen Self-Organizing Map (SOM) can be defined as a competitive unsupervised learning neural network [4]. When an observation is recognized, the activation of an output cell - competition layer - inhibits the activation of other neurons and reinforce itself. It is said that it follows the so called "Winner Takes All" rule. Actually, neurons are specialized in the recognition of one kind of observation. A SOM consists in a two dimensional map of neurons. Each neuron j is connected to n inputs according to n weights connections (the prototype vector) $w^j = (w_0^j, ..., w_n^j)$ and to its neighbors with topological links. The training set is used to organize these maps under topological constraints of the input space. Thus, a mapping between the input space and the network space is constructed; two close observations in the input space would activate two close neurons of the SOM. Usually a radial symmetric Gaussian neighborhood function K_{ij} is used for this purpose.

The basis algorithm of our approach is the KDisj method proposed in [5]. This algorithm is an adaptation of SOM that allow to project on the map both data and features used to describe them. This algorithm is designed for the quantification of qualitative data in the form of a disjunctive table D: each feature has several mutually exclusive modalities (e.g. the attribute "color" may have the modalities "yellow", "green", etc ...). Features can therefore be encoded as a vector size equal to the number of modalities with a value of zero ??in all dimensions except one. We can code in the same way several attributes by a vector of size equal to all the modalities of the various features with as many

non-zero values ??as the number of attributes. The main idea of KDisj is that one can describe a data based on the modalities associated with (row vector), but it is also possible to describe a modality based on the set of data (column vector). All data and modality can then be represented in a space of dimension $A + E$ (number of modalities for all features + number of data). A SOM can be learned in this space by presenting alternately a data and a modality during the learning. The distance between a data (size A) and a prototype of the map (size $A + E$) will be calculated on the A first dimensions, while the distance between a modality (size E) and a prototype will be calculated on the last E dimensions. To ensure a link between the A first dimensions and the E last, prototypes will be adjusted on all dimensions during the adaptation phase, by associating to each data its not-null modality the most characteristic (i.e. the rarest in the data set). Thus, the first A dimensions of each prototype are adapted based on the presented data and the last E dimensions are adapted depending on the associated modality. Note that it is not possible to do this even when a modality is presented, since there is no rare data in the description of the set of modalities (each data is characteristic of exactly as many modalities as the number of attributes).

3 A New Two-Levels Coclustering Algorithm: S2L-KDisj

The key idea of the two-level clustering approach based on SOM is to combine the dimension reduction and the fast learning capabilities of SOM in the first level to construct a new reduced vector space, then applies an other clustering method in this new space to produce a final set of clusters in the second level [7,8]. Although the two-levels methods are more interesting than the traditional approaches (in particular by reducing the computational time and by allowing a visual interpretation of the partition result), the data segmentation obtained from the SOM is not optimal, since a part of information is lost during the first stage (dimension reduction). Moreover, this separation in two stages is not suited for a dynamic (incremental) segmentation of data which move in time, in spite of important needs for analysis tools for this type of data.

3.1 Principle of the Algorithm

We propose here a new unsupervised learning algorithm, Simultaneous Two-Levels - KDisj (S2L-KDisj), which learns simultaneously the structure of the data and its segmentation (section 2). In S2L-KDisj, we propose to associate each neighborhood connection a real value ν which indicates the relevance of the connected neurons. Given the organization constraint of the SOM, both best closet prototypes of each data must be connected by a topological connection. This connection “will be rewarded” by an increase of its value, whereas all other connections from the winner neuron “are punished” by a reduction of their values. Thus, at the end of the training, a whole of inter-connected prototypes will be an artificial image of the relevant sub-group of the whole data set.

3.2 Algorithm

The connectionist learning is often presented as a minimization of a cost function. In our case, it will be carried out by the minimization of the distance between the input samples and the map prototypes, weighted by a neighborhood function K_{ij}. To do that, we use a gradient algorithm. The cost function to be minimized by the algorithm is:

$$\tilde{R_z}(w) = \frac{1}{N}\sum_{k=1}^{N}\sum_{j=1}^{M} K_{jN(z^{(k)})} \parallel w^j - z^{(k)} \parallel^2 \quad \text{with} \quad K_{ij} = \frac{1}{\lambda(t)} \times e^{-\frac{d_1^2(i,j)}{\lambda^2(t)}}$$

N represents the number of learning samples, M the number of neurons in the map, $N(z^{(k)})$ is the neuron having the closest weight vector to the input $z^{(k)}$ (note that $z^{(k)}$ could be either a data or a feature), and K_{ij} is a positive symmetric kernel function: the neighborhood function. $d_1(i,j)$ is the Manhattan distance defined between two neurons i and j on the map grid. $\lambda(t)$ is the temperature function modeling the topological neighborhood extent, defined as $\lambda(t) = \lambda_i(\frac{\lambda_f}{\lambda_i})^{\frac{t}{t_{max}}}$. λ_i and λ_f being respectively the initial and the final temperature. t_{max} is the maximum number of iterations.

The S2L-KDisj training process is inspired from the SOM adaptation proposed in [9]. Whenever a data or a modality is presented, the value of the connection between the two most representative prototypes is increased whereas other connections values are decreased. The version presented here is modified to be adapted to data expressed in frequency or proportion, i.e. we associate a percentage to each modality of a feature, the sum of terms for an attribute is thus equal to 1 (or 100%). This data type is widely used in many fields (time management, budget, modality varying in time or space,...). The only difference with a disjunctive table, in this case, is that you can associate a characteristic data to each modality. This allow to update prototypes in all dimensions (A+E) whatever is presented (data or modality).

The S2L-KDisj algorithm is the following:

1. **Initialization:**
   - Correct disjunctive table D into D_c:

$$d_{ij}^c = \frac{d_{ij}}{\sqrt{d_{i.}d_{.j}}} \text{ with } d_{i.} = \sum_{j=1}^{N} d_{ij} \text{ and } d_{.j} = \sum_{i=1}^{N} d_{ij}$$

     In that way using euclidean distance on D_c is similar to use weighted χ^2 distance on D [5].
   - Initialize randomly the prototypes $w^j = (w^{Aj}, w^{Ej})$.
   - Initialize to 0 connections values ν_{ij} between each pair of neurons i et j.

2. **Present a data** $x^{(k)}$: i.e. a row of D_c, randomly chosen.
   - Associate to $x^{(k)}$ modality $y(x^{(k)})$ defined by
   $$y(x^{(k)}) = \underset{y}{Argmax}\, d^c_{xy}$$
   and create vector $Z_x^{(k)} = (x^{(k)}, y(x^{(k)}))$.
   - *Competition step* :
     - Choose the two most representatives neurons $u^*(x^{(k)})$ and $u^{**}(x^{(k)})$ over the A first dimensions:
     $$u^*(x^{(k)}) = \underset{1\leq i\leq M}{Argmin} \parallel x^{(k)} - w^{Ai} \parallel^2$$
     $$u^{**}(x^{(k)}) = \underset{1\leq i\leq M, i\neq u^*}{Argmin} \parallel x^{(k)} - w^{Ai} \parallel^2$$
     - Update connection value between $u^*(x^{(k)})$ and its neighbors according to the learning step $\varepsilon(t)$, a decreasing function of time between $[0, 1]$, inversely proportional to time:
     $$\nu_{u^*u^{**}}(t) = \nu_{u^*u^{**}}(t-1) - \varepsilon(t)\,(\nu_{u^*u^{**}}(t-1) - 1)$$
     $$\nu_{u^*i}(t) = \nu_{u^*i}(t-1) - \varepsilon(t)\,(\nu_{u^*i}(t-1)) \quad \forall i \neq u^{**}, i \text{ neighbor of } u^*$$
   - *Adaptation step* :
     - Update prototypes w^j for each neuron j on all dimensions :
     $$w^j(t) = w^j(t-1) - \varepsilon(t) K_{ju^*(x^{(k)})}(w^j(t-1) - Z_x^{(k)})$$

3. **Present a modality** $y^{(k)}$: i.e. a column of D_c, randomly chosen.
   - Associate to $y^{(k)}$ modality $x(y^{(k)})$ defined by
   $$x(y^{(k)}) = \underset{x}{Argmax}\, d^c_{xy}$$
   and create vector $Z_y^{(k)} = (x(y^{(k)}), y^{(k)})$.
   - *Competition step* :
     - Find the two best representatives neurons $u^*(y^{(k)})$ and $u^{**}(y^{(k)})$ over the E last dimensions and update connection values between $u^*(y^{(k)})$ and its neighbors as in step 2.
   - *Adaptation step* :
     - Update prototypes w^j for each neuron j :
     $$w^j(t) = w^j(t-1) - \varepsilon(t) K_{ju^*(y^{(k)})}(w^j(t-1) - Z_y^{(k)})$$

4. **Repeat steps 2 and 3** until convergence.

At the end of the clustering process, a cluster is a set of prototypes which are linked together by neighborhood connections with positive values. Thus, the right number of cluster is determined automatically.

4 Application

The application part of this work is to analyze and visualize biological experimental data. These data comes from a study on the ants' spatial and social organization [10]. A queen (R), a male (Mc), a young (J) and 43 workers (2-44) were observed in an artificial nest composed of 9 rooms (Loc2 to Loc10), a tunnel leading outside (Loc1) and a foraging area (Loc0, see Figure 1). For each individual, we know the proportion of time spent in each room and in 20 different activities extracted from a set of pictures of all individuals in the nest and the foraging area.

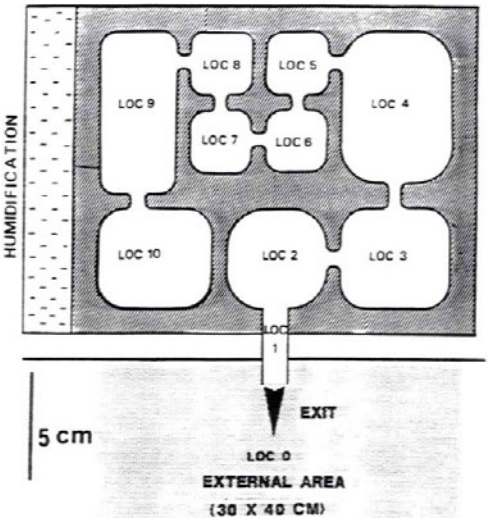


Fig. 1. The artificial nest used for the experimental study

The main goal of this study is to determine the existence of clusters of similar ants and to link each group of ants with some characteristic behaviors, in order to understand the social role of the group, as well as the relevant location, in order to understand how each group manage the allocated space to perform its task. S2L-KDisj is then a relevant algorithm to perform these tasks, as it is able to produce cluster regrouping at the same time individuals and features modalities. In addition, the visualization capability of the SOM algorithm can be used to visualize the resulting coclustering.

The results obtained with S2L-KDISJ from these data are shown in Figure 2. The entire learning process took a few seconds. Codes C0 to C10 represent the ten rooms. Ants behaviors are represented by 20 activities, each coded with a two or three letters, the last one giving the general category (T: entry and exit of the nest, N: Management of food, C: cocoons care, L: larvae care, O: eggs care).

These results show that the queen, the young and a few other individuals are related to Room 9 and are characterized by "immobility on eggs and larvae"

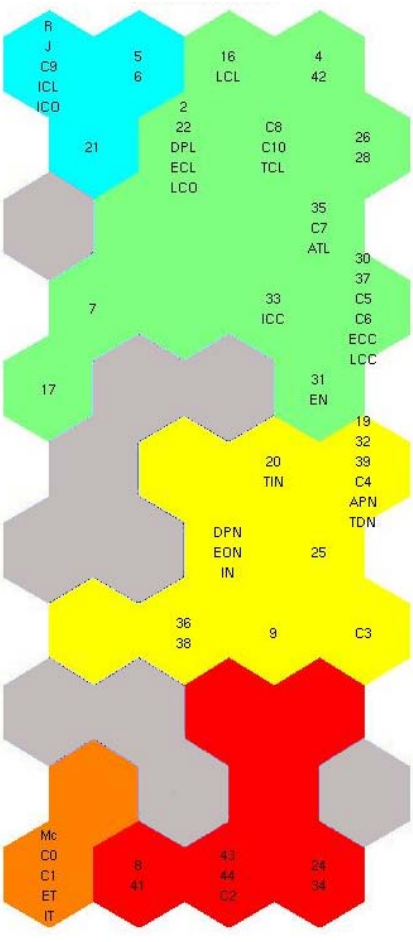


Fig. 2. Clusters of ants (numbers), behaviors (letters) and location (C + number) obtained automatically with S2L-KDisj. Each hexagon is a visualization of a neuron of the map. Neurons sharing a color represent individuals and features belonging to the same cluster. Grey neurons are not representative and do not belong to any clusters.

behavior ("blue" cluster). This is relevant as the queen need to be in a big room far from the entrance (for protection, [11]). Also, as the queen spend her life to lay eggs, there is always in eggs and sometime larvae in her room, as well as young ants that don't have any social activity yet [11]. The "green" cluster regroup rooms 5, 6, 7, 8 and 10 with activity of larvae and cocoons care. This group is representative of the social role "nurses" which is essential in the colony's life. Ants in this group take care of the brood in order to guarantying its survival. As during the development of the larva and the cocoon the need in humidity and temperature may vary, it have been observed hat the nurses displace frequently the members of the brood to find optimal location [11], it is

therefore not surprising to find many different rooms in this cluster. In the same way, cluster "yellow" is a group of ant managing food in room 3 and 4, not far from the foraging area (where the food is given). The "red" cluster represent ants spending most of their time in room 2, without any related social activities. These kind of ant are known to be "generalist" in a colony, they are able to perform any task, especially foraging task, depending on the need of the colony [10]. The last cluster ("Orange") regroup rooms 0 and 1 (the tunnel and the foraging area) with input and output behavior. Theses relations are obvious. The male is also in the cluster, which indicate that he is mature to flight out the nest to find a female and fund a new colony.

One should also note that the linear disposition of the rooms inside the nest is also kept on the map in S2L-KDisj.

5 Conclusion

In this paper, a new coclustering algorithm is proposed, based on the learning of a SOM. The advantages of this algorithm is to combine the speed and the visualization capability of the SOM with the possibility to perform a coclustering analysis that detect automatically the number of clusters to find. We applied this algorithm to analyze characteristics of spatial and social organization in an ant colony. Obtained results are easy to read and understand, and are perfectly compatible with biologists knowledge. Thus, this tools is a good candidate to be used in experimental research in Biology.

References

1. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall, Inc., Upper Saddle River (1988)
2. Mirkin, B.: Mathematical Classification and Clustering. Nonconvex Optimization and Its Application, vol. 11 (1996)
3. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. Trans. on Computational Biology and Bioinformatics 1(1), 24–45 (2004)
4. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (2001)
5. Cottrell, M., Letrémy, P., Roy, E.: Analysing a contingency table with kohonen maps: A factorial correspondence analysis. In: Mira, J., Cabestany, J., Prieto, A.G. (eds.) IWANN 1993. LNCS, vol. 686, pp. 305–311. Springer, Heidelberg (1993)
6. Hoang, T., Olteanu, M.: SOM biclustering – coupled self-organizing maps for the biclustering of microarray data. In: IDAMAP 2003, Workshop Notes, pp. 40–46 (2003)
7. Hussin, M.F., Kamel, M.S., Nagi, M.H.: An efficient two-level SOMART document clustering through dimensionality reduction. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) ICONIP 2004. LNCS, vol. 3316, pp. 158–165. Springer, Heidelberg (2004)
8. Ultsch, A.: Clustering with SOM: U*C. In: Workshop on SOM, pp. 75–82 (2005)
9. Cabanes, G., Bennani, Y.: A simultaneous two-level clustering algorithm for automatic model selection. In: ICMLA 2007, pp. 316–321 (2007)
10. Fresneau, D.: Biologie et comportement social d'une fourmis ponérine néotropicale (*Pachycondyla apicalis*). PhD thesis, Université Paris-Nord (Paris 13), Paris (1994)
11. Hölldobler, B., Wilson, E.: The ants. Harvard University Press (1990)

An Evolutionary Fuzzy Clustering with Minkowski Distances

Vivek Srivastava, Bipin K. Tripathi, and Vinay K. Pathak

Harcourt Butler Technological Institute, Kanpur, India
{viveksrivastavakash,abkt.iitk,vinaypathak.hbti}@gmail.com

Abstract. In this paper, we present a novel evolutionary fuzzy clustering approach with Minkowski distances. Fuzzy clustering plays an important role for various kinds of classification problems. Evolutionary algorithm is used for searching the best partitioning among the populations generated by different runs of the fuzzy clustering algorithm. Evolutionary fuzzy clustering performs better as compared to the conventional fuzzy clustering in terms of classification accuracy and partitioning. Fuzzy c-means (FCM) is a data clustering algorithm in which each data point is associated with a cluster through a membership degree. Here, Minkowski distance is used with FCM instead of conventional Euclidian distance because of its more generalized nature. It does not restrict the shape of the clusters generated. Empirical evaluation demonstrates the performance of proposed novel technique in terms of precision and accuracy in various benchmark problems.

Keywords: Evolutionary Algorithm, Fuzzy Clustering, Minkowski Distances.

1 Introduction

Clustering is defined as grouping of data based on their similarities in some context. Clustering is deemed as one of the most difficult and challenging area in machine learning, because of its unsupervised nature. One distinguishes hard and soft types of clustering. Hard clustering deals with grouping of data into number of clusters having crisp boundaries while soft clustering deals with fuzzy boundaries in which each object belongs to one or more clusters to different degree [1]. Fuzzy clustering has been widely applied in substantive areas such as business, medical, engineering, bioinformatics, image processing problems. In classification and pattern recognition problems, fuzzy clustering shows better scope [2] with combination of neural network [3]. Evolutionary computation is inspired by the biologically phenomenon of human evolution and it is one of the emerging field in computational intelligence. In recent researches, evolutionary algorithms are being used with fuzzy clustering for better classification and identification [4] [5]. The main motivation behind combination of evolutionary computation with fuzzy clustering is optimal search to yield more accurate solution. Most of the research on evolutionary algorithm for fuzzy clustering is based on the evolution of fuzzy partition of data when the number of cluster is given or fixed [5]-[7]. In case of unknown number of clusters, the recent papers adopt to optimize both numbers of clusters and corresponding fuzzy partitions with the help

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 753–760, 2011.
© Springer-Verlag Berlin Heidelberg 2011

of evolutionary search [8]. It is observed that the fuzzy clustering algorithm, generally speaking FCM algorithm is used with Euclidian distance as the dissimilarity measure. In [13], FCM is used with wavelet transform and recognition is done based on Euclidian distance measure and correlation coefficient for human face recognition. FCM is used as distribution of training images to number of clusters and then these patterns are fed into the parallel neural network to obtain the final recognition in [2]. In this paper, we propose an approach which incorporates evolutionary search with fuzzy clustering using minkowski distance. The paper is organized as follows: Section 2 deals with the proposed evolutionary fuzzy clustering with minkowski distances and section 3 shows experimental results and analysis. Section 4 is devoted to conclusion and future scope.

2 Evolutionary Fuzzy Clustering with Minkowski Distances

Evolutionary searching is generally used for finding the best possible solution among the existing ones. In fuzzy clustering, initial partitioning is created randomly which satisfies (1), (2) and (3), therefore different runs of the same algorithm may produce different partitions of the same dataset [4]. Objective function also plays a vital role for proper partitioning. Objective function describes the inter classes similarties which is minimized in each iteration. Various kinds of evolutionary algorithms have been proposed for overcoming these limitations such as multi objective evolutionary clustering [11] and ensemble based evolutionary clustering [12]. We propose an evolutionary search approach for finding the best population among the various off springs generated by the different runs of the same algorithm. We call this technique as evolutionary fuzzy clustering with minkowski distances (EFC-MD). EFC-MD involves the evolutionary search approach on different runs of the Fuzzy clustering in which minkowski distance is used. In this section we define steps of this proposed technique as the membership function, chromosome representation, population initialization, computation of fitness function and selection process of best population.

2.1 Membership Function

Let X= { $x_1, x_2, x_3, \ldots\ldots x_N$ } is input dataset having each elements of *n*-dimensions. Fuzzy c-mean clustering algorithm divides N datasets to C clusters with fuzzy partition matrix U of size C×N which is known as membership function.

Membership function is defined as U = [μ_{ik}] is C×N matrix which satisfied the following constraints –

$$\mu_{ik} \in [0,1] \quad \forall\, 1 \le i \le C \qquad \text{And} \quad 1 \le k \le N \tag{1}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N \quad \forall\, 1 \le i \le C \tag{2}$$

$$0 < \sum_{i=1}^{C} \mu_{ik} = 1 \qquad \forall 1 \le k \le N \quad \text{Where} \quad 2 \le C \le N \tag{3}$$

2.2 Chromosome Representation

Each chromosome is a sequence of attribute values representing C clusters. Let $\Theta = \{C_{ij}\}$ where C_{ij} is defines as

C_{ij} = { 1 if jth data set belongs to ith cluster, 0 Otherwise }

Where $1 \le i \le C$ and $1 \le j \le N$

2.3 Population Initialization

Initially C clusters are encoded in each chromosome and population is initialized randomly .Therefore in each run different initial population is generated.

2.4 Computation of Fitness Function

The Objective function for EFC-MD is defined as following

$$J(\mu, O) = \sum_{k=1}^{N} \sum_{i-1}^{C} (\mu_{ik})^m d^{2\beta}(x_{k,} O_i) \quad ,$$

$$d^{\beta}(x_{k,} O_i) = \left(\sum_{j=1}^{n} \| x_{kj} - O_{ij} \|^p \right)^{\beta / p} \quad ,1 \le p < \infty, \quad 0 < \beta \le 1 \qquad (4)$$

Where m is a weighting exponent which is known as fuzzifier. Generally, the value of m lies between one to infinity. Value of m greatly influences the performance of FCM algorithm. When m approaches to infinity, the solution will be the center of gravity of whole dataset and when m=1 it behaves like classical c mean. There fore selection of suitable fuzzifier m is very important for implementation of FCM. In [10], it has been shown that a proper weighting exponent value depends on data itself. Parameter p is the order of the distance measure which behaves as Euclidian distance for p=2. $d^{\beta}(x_{k,} O_i)$ is minkowski distance and in objective function we use squared minkowski distance. Main motivation behind using minkowski distance in stead of using Euclidian distance is that the shape of the clusters decided for the given problem mainly depends upon the distance measure taken. The exact nature of these parameters depends on the shape of clusters to be generated, which may be boxes, ellipsoids, spheres and others. Selection of this distance measure does not tend the shape of cluster spherical which is often in Euclidian distance. The introduction of power β allows controlling the loss function against outliers [9]. In [9], it has been shown that for $p =$ 1 and β=0.5 and $p =$ 2 and β=1.0, the results are robust. We have tested EFC-MD for different values of p and the best results have been obtained for the experimental data used in this paper at p=4. In fuzzy clustering, we minimize the objective function which means the fitness function is inversely proportional to the objective function. Hence

Fitness function $$f = \left[\frac{1}{\sum_{k=1}^{N} \sum_{i-1}^{C} (\mu_{ik})^m d^{2\beta}(x_{k,} O_i)} \right] \qquad (5)$$

Higher value of *f* gives survival to the fittest population and best population is selected among the various offsprings generated on different runs.

EFC-MD algorithm is iterated through the necessary conditions for minimizing the objective function with the following updates in member function and centre of the clusters:

$$O_i = \frac{\sum_{k=1}^{N} \mu_{ik}^{m} x_k}{\sum_{k=1}^{N} \mu_{ik}^{m}} \tag{6}$$

$$\mu_{ik} = \frac{d(x_k, O_i)^{\frac{1}{m-1}}}{\sum_{j=1}^{C} d(x_k, O_i)^{\frac{1}{m-1}}} \qquad i = 1,2.....,C \tag{7}$$

Let $J(\mu, O)^{(t)}$ is the objective function at *t*th iteration then The EFC-MD algorithm termination condition is as following-

$\| J(\mu, O)^{(t+1)} - J(\mu, O)^{(t)} \| < \Im$ Where $\Im$ is the pre specified threshold. (8)

2.5 Selection

Let $P^{(0)}, P^{(1)}, ...P^{(k)}$ be the *k* populations generated by the k iterations of EFC-MD algorithm. For *k=1,….,t* where *t* is the total number of runs generate the offsprings $P^{(k+t)}$ using (4) and termination criteria (8). Selection is done based on the elitism function which selects the best chromosome first.

EFC-MD iteratively finds the best population among the various population generated by different runs. After getting the best partitioning for training sets, testing is performed by matching the unknown test data with the centre of the clusters generated.

3 Experimental Results and Analysis

In order to evaluate the performance of proposed technique, empirical evaluation is carried out over three benchmark problems [14, 16] Fisher's Iris data set, wine data set and SPECTF heart dataset problems. The results obtained were compared with the ones shown by other methodology. The approximation capability has been shown in terms of number of misclassifications for training data (training error) and testing data (testing error).

3.1 Iris Data Problem

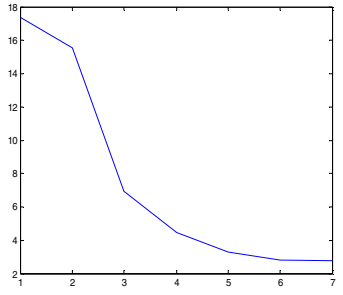
This is a well-known three-class benchmark problem. This data set contains three different classes of flowers with four attributes. The dataset in [14] provide 150

instances of all three classes, training is performed on 75 data values and testing is performed on other 75 data values selected randomly. Chromosome representation is shown in Table 1. Best partitioning in training data set is obtained by selecting that population having higher fitness value using (5). Table 3 shows that increase in value of m gives lower recognition rate when we use fuzzy clustering algorithm with Euclidian distance while on the other hand proposed EFC-MD gives quite consistent result and higher classification accuracy.

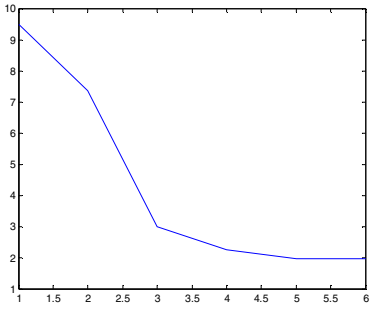
Table 1. Chromosome representation for 75 training Iris dataset

	1	……	25	26	27	……	50	51	…..	75
C1	1	……	1	0	0	…….	0	0	……	0
C2	0	…….	0	1	1	……	1	0	…….	0
C3	0	…….	0	0	0	…….	0	1	…….	1

For the same training and testing dataset, EFC-MD gives better results as compared with traditional fuzzy clustering. When we use 75 dataset as training set, we get best results on m=1.25 , p= 4 , $\Im$ =0.01 and β =1.0. Average number of iterations using EFC-MD is quite less than the iterations used in fuzzy clustering.



(a)



(b)

Fig. 1. Plot between objective function and number of iteration (a) Using fuzzy clustering at m=1.25 and $\Im$ =0.01 (b) Using EFC-MD at at p= 4 , $\Im$ =0.01, β =1.0 for Iris data problem

It has been observed that on increasing the value of weighting exponent *m*, we get lower classification accuracy. In most of the cases value chosen greater than 3 degrades recognition rate. Therefore the best classification accuracy is obtained at lower values of *m*. On training, we get almost 100% classification accuracy on EFC-MD while in testing we obtain 96% classification accuracy. Table 2 shows the final cluster centers for all four dimensions.

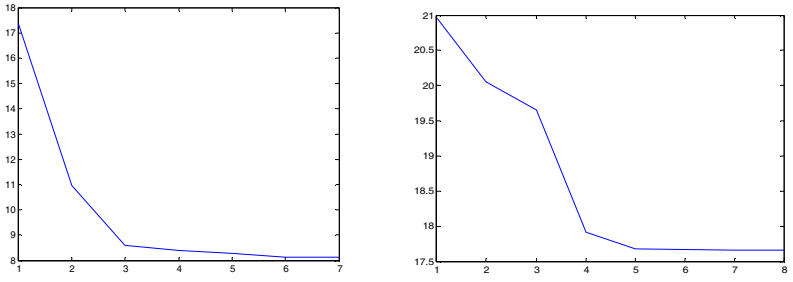
Table 2. Final values of 3 cluster centers for Iris data

C1	0.2618	0.5933	0.1624	0.1493
C2	0.4788	0.3581	0.5493	0.5155
C3	0.6101	0.4110	0.7310	0.7493

Above result shows the Evolutionary fuzzy clustering performs better as compared with traditional FCM because for the same data set, it gives more classification accuracy and more accurate partitioning which are very close to either 0 or 1. Final value of objective function $J(\mu,O)$ = 1.954132 using EFC-MD and $J(\mu,O)$ = 1.966103 when we are using traditional fuzzy clustering. It is clear than EFC-MD optimizes objective function more closely as compared to other related technique.

3.2 Wine Data Problem

The wine data set is the outcome of the chemical analysis of wine based on 13 constituents varies in three different kinds of wine classes. There are total of 178 data values of all three classes. In this experiment, we considered 58% of data for training and 42 % data for testing. We get the best results for *p*= 4, $\Im$ =0.01, β =1.0. In this case training error is only 1.94% and 6.66% testing error. Overall classification accuracy is 93.4%.



(a) (b)

Fig. 2. Plot between objective function and number of iteration Using EFC-MD at *p*= 4, $\Im$ =0.01, β =1.0 and *m*=1.05 for (a)wine data (b) SPECTF heart data

We get final value of $J(\mu, O)$ = 8.117299. After obtaining the final results , we chose to carry out cross validation 10 different combinations of training and testing sets and averaging the results of accuracy , we get finally 96% accuracy in this case. On comparison with conventional FCM and EFC-MD using same parameters, we get 46 misclassifications out of 103 data values. Due to scarcity of space in this paper, we are not providing the related table which shows the cluster centre values for wine data set and SPECTF heart dataset, but itself explanatory for readers as given in previous example.

Table 3. Comparision of classification errors on various values of fuzzifier `*m*' in terms of Traing error (Tr) , and Testing error (Ts) for three different benchmark problems considered

m	FCM at $\Im$ =0.01						EFC-MD at *p*= 4, $\Im$ =0.01, β =1.0					
	Iris		Wine		SPECTF		Iris		Wine		SPECTF	
	Tr	Ts	Tr	Ts	Tr	Ts	Tr	Ts	Tr	Ts	Tr	Ts
1.05	0	4	34	46	0	60	0	3	2	5	0	27
1.1	0	4	34	46	0	60	0	3	2	5	0	27
1.2	0	5	34	46	0	60	0	3	2	5	0	27
1.25	0	5	34	46	0	60	0	3	2	5	0	27

3.3 SPECTF Heart Problem

This data set [16] is based on cardiac single proton emission computed tomography (SPECT) images. Each patient is classified into two categories normal and abnormal. Dataset contains 267 instances each of them having 44 attributes. In [16], it is recommended to take 80 instances for training and 187 instances for testing out of 267 instances. Using EFC-MD, we get 100% accuracy for training data and 85.56% accuracy in testing data. Final value of $J(\mu, O)$ =17.663090 is obtained at, at *p*= 4, $\Im$ =0.001, β =1.0 and *m*= 1.06. Accuracy in this case is 85.56% which is more than the accuracy of 77% using CLIP3 algorithm [15] for SPECTF dataset.

Therefore, it has been thoroughly observed that EFC-MD performs far better than conventional FCM. Table 3 shows the comparatively analysis of FCM with EFC-MD. It has also been seen that the proposed technique gives quite consistent results when we chose value of *m* in range of 1 to 1.25. Further increase in *m* from this limit increases the number of misclassifications in our experiments.

4 Conclusions

This paper presents a novel evolutionary fuzzy clustering technique with minkowski distances. The idea is based on the combination of evolutionary search for finding the best partitioning with fuzzy c-mean clustering using minkowski distances implied in the given dataset. It outperforms to conventional fuzzy clustering and yield higher

accuracy. It also gives results with better precision in partitioning. Proposed technique shows its capability in terms of finding best partitioning efficiently. In all three experiments, it has been shown that EFC-MD performs better at lower values of fuzzifier.

Acknowledgments. Present work is supported by the department of computer science and engineering, H.B.T.I.,Kanpur and G.B.T.U., Lucknow.

References

1. Hoppner, F., Klawonn, F., Kruse, R., Runker, T.: Fuzzy cluster analysis: methods for classification, data analysis and image recognition. Wiley, New York (1999)
2. Lu, J., Yuan, X., Yahagi, T.: A method of face recognition based on fuzzy c means clustering and associated sub NNs. IEEE Trans. on Neural Network 18(1), 150–160 (2007)
3. Tripathi, B.K., Kalra, P.K.: On efficient learning machine with root-power mean neuron in complex-domain. IEEE Transaction on Neural Network 22(5), 727–738 (2011)
4. Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., Carvelho, A.C.P.: A survey of evolutionary algorithms for clustering. IEEE Trans. on System Man and Cybernetics-Part C: Applications and Reviews 39(2), 133–155 (2009)
5. Le, T.V.: Evolutionary fuzzy clustering. In: Proc. IEEE Congr. Evol. Comput., pp. 753–758 (1995)
6. Hall, L.O., Ozyurt, I.B., Bezdek, J.C.: Clustering with genetically optimized approach. IEEE Trans. Evol. Comput. 11(2), 103–112 (1999)
7. Yuan, B., Klir, G.J., Stone, J.F.S.: Evolutionary Fuzzy c mean clustering algorithm. In: Proc. Int. Conf. Fuzzy Syst., pp. 2221–2226 (1995)
8. Fazendeiro, P., Oliveira, J.V.: A semantic driven evolutive fuzzy clustering algorithm. In: Proc. IEEE Int. Conf. Fuzzy Syst., pp. 1–6 (2007)
9. Groenen, P.J.F., Jajuga, K.: Fuzzy clsuetring with squared minkowski distances, fuzzy sets and systems, vol. 120, pp. 227–237 (2001)
10. Yu, J., Cheng, Q., Hung, H.: Analysis of weighting exponent in the FCM. IEEE Trans. on Syst. Man, Cybern. B, Cybern. 34(1), 634–638 (2004)
11. Deb, K.: Multi-Objective Optimization using evolutionary algorithms. Wiley, New york (2001)
12. Handl, J., Knowles, J.: An evolutionary approach to multiobjective clustering. IEEE Trans. on Evol. Comput. 11(1), 56–76 (2007)
13. Yoon, C., Park, J., Park, M.: Face recognition using wavelet and fuzzy c-means clustering. In: IEEE Int. Conf. Tencon (1999)
14. Iris, Wine Dataset, `http://archive.ics.uci.edu/ml/datasets`
15. Cios, K.J., Wedding, D.K., Liu, N.: CLIP3: cover learning using integer programming. Kybernetes 26(4-5), 513–536 (1997)
16. SPECTF Heart Dataset, `http://archive.ics.uci.edu/ml/datasets/SPECTF+Heart`

A Dynamic Unsupervised Laterally Connected Neural Network Architecture for Integrative Pattern Discovery

Asanka Fonseka[1], Damminda Alahakoon[1], and Jayantha Rajapakse[2]

Cognitive and Connectionist Systems Lab
[1] Faculty of Information Technology, Monash University,
Clayton, Australia
[2] Sunway Campus, Monash University, Malaysia
{asanka.fonseka,damminda.alahakoon,
jayantha.rajapakse}@monash.edu

Abstract. We describe an unsupervised neural network approach to build associations between neurons within cortical maps. These associations are then used to capture patterns in the input data. The cortical maps are modeled using growing self-organization maps to capture the input stimuli distribution within a two dimensional neuronal map. The associations are modeled using passive lateral connections using recognition frequency of input stimuli by a neuron. The proposed approach introduces a novel way of learning by adapting neighborhood learning rules and proximity measures according to the input stimuli structure.

Keywords: neural networks, cross-modal, lateral connection, association discovery, self-organizing maps.

1 Introduction

Most decisions require the integration or fusion of information from multiple sources. These sources represent information about the same objects or events from different viewpoints. The nature of these sources can be multimodal or pertain to the same modality. Majority of state of the art artificial fusion techniques follow "post-perceptual" integration phenomena [1-3] where modalities or multiple cues associated with the same modality are treated separately and processed. Finally, the outputs of these individual channels are merged together in order to generate the global decision or perception. The main limitation associated with this strategy is that the unimodal perception does not account for inter-sensory influence.

As opposed to the *post-perceptual* integration the *cross-modal-influence* strategy shares information across the stimulated modalities during perceptual binding. A very few unsupervised neural network techniques which inspired by the cross-modal integration have been proposed [4-6] including some theoretical frameworks [7,8]. Most of these techniques are model based and incapable of offering specialized attention to the incoming input stimuli structure. Also the learning of these neural networks does not reflect the input stimuli distribution over time and incapable of handling multimodal data.

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 761–770, 2011.
© Springer-Verlag Berlin Heidelberg 2011

It is important to develop effective approaches in integrative pattern discovery and knowledge mining. Integrative analyses can reduce the bias, error and ultimately generate better, accurate results compared to individual data source analysis.

The main contribution of this paper is the development of a dynamic unsupervised neural network framework for integrative pattern discovery in a more meaningful manner which resembles the cross-modal-influence phenomena in perceptual binding. The proposed architecture can also be used to explain the phenomenon "binding problem". The model addresses the issues highlighted above in the existing techniques and further it fosters unsupervised fusion techniques by introducing a novel biologically motivated approach, since not many research attempts have been devoted to this area.

Cortical modeling of the proposed neural network approach is accomplished by employing the Growing Self-Organizing Map (GSOM) [9] which has a dynamic self organizing architecture. Allocation of neurons within the growing self-organizing map is governed by the input stimuli received during the growing phase of the learning. The cross-modal-influence is implemented across different modality channels using excitatory and inhibitory lateral connections. Synaptic weights of these lateral connections are adapted according to the Hebbian rule [10] by considering the input stimuli distribution (over time) of the receptive fields. Once the network completes learning for given multi source stimuli, correlation and interdependencies between the learned prototypes can be obtained using the synaptic weights of the lateral connections. The proposed architecture is not only designed for processing static input stimuli, but also capable of processing temporal sequences by utilizing the *Dynamic Time Warping* (DTW) approach. This exhibits the capability of handling various data types and structures under a unified framework. Further we argue that the neighborhood synapses learning rule should be specialized to different modalities in order to facilitate the behavior of the underlying input stimuli structure.

The proposed neural network architecture can be applied to multimodal data sources as well as to multiple data sources of a single modality. Experiments were designed to evaluate the performance and validity of the proposed neural network approach by exploiting multimodal data sources and multiple data sources pertaining to a single modality. In order to keep the implementation details of this framework simple we only consider two cortical areas or two multi modal channels. Without loss of generality it is logical to generalize this approach for multiple cortical areas.

2 Method

2.1 Model Overview and Architecture

The proposed neural network model is composed of two cortical areas modeled by growing self organizing maps. These two cortical areas may represent modality specific neurons belonging to two modalities or different cortical areas responding to two types of specific input stimuli belonging to the same modality. The maps are organized in a way such that proximal neurons respond to proximal input stimulus enabling topology preservation. To begin learning, each map is initialized with four neurons and its structure changes over time (usually referred to as *growing phase*) by

appending more neurons to the map in order to capture the underlying input stimuli distribution within the map [9]. A single neuron in the map can be considered the convergence point of the underlying neurons of its receptive field. In our model we assume that neurons' receptive fields represent input stimuli appearing from sensory channels. Each neuron in a map is not only connected with afferent input stimuli, but it is laterally connected to all the neurons in the other map. These lateral connections are passive and do not influence the neuronal dynamics. Hence these lateral connections are considered *passive lateral connections*. A general sketch of the model is depicted in Fig. 1. The passive lateral connections are learned according to the Hebb rule which is often interpreted as "strengthen the connection between neurons that tend to be active at the same time" [10].

Once the two cortical maps are trained for given input stimuli the resultant neurons in the maps act as concepts or prototypes which generalize their input stimuli. The passive lateral connections can be interpreted as concept associations and the connection strength of the passive lateral connections reflect the frequency of the input stimuli present at the receptive fields.

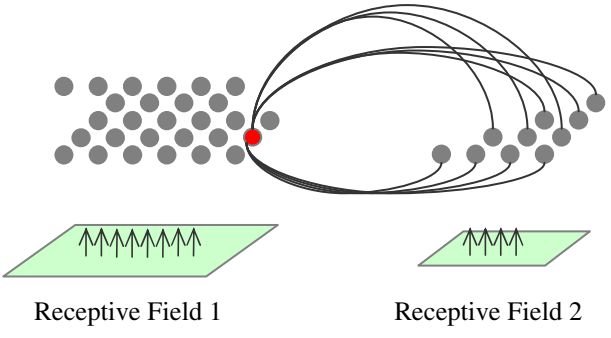


Fig. 1. Architecture of the proposed neural network. The red circle denotes the wining neuron and it is connected to all other neurons in the second map. Connections from the right side map to the left side map are not shown.

2.2 Algorithm

Mathematical Description. There are two operations which form the basis for various self-organizing processes, and are common to both the growing and smoothing phases of the GSOM learning. The first operation locates the best matching node. The second operation smooth out the weight values in the best matching units and its topological neighbors.

Let C^k be any cortical map which converges with the inputs from any sensory channel k. Let $\vec{E}$ be any sequence of input stimuli or object instances whose feature space is characterized by k sensory stimuli ($|\vec{E}| = M$). Any input stimulus $\overrightarrow{e_i} \in \vec{E}$ is composed of sensory inputs $\overrightarrow{x_i^k} \in \mathbb{R}^{N_k}$. Each C^k map will be initialized with four neurons and each neuron j is associated with a weight vector $\overrightarrow{\vartheta_j^k} \in \mathbb{R}^{N_k}$, frequency vector $\overrightarrow{\tau_j^k} \in \mathbb{R}^M$, two dimensional coordinate $\vec{r} \in \mathbb{Z}^2$ and hit counter $\lambda \in \mathbb{N}^0$. The

weight vectors are initialized to random values and frequency vectors are initialize to zero. For an input $\overrightarrow{x_i^k}$ there will be a winner neuron ω^k satisfying,

$$\omega^k = \underset{j \in C^k}{\arg min}\, D^k(\overrightarrow{x_i^k}, \overrightarrow{\vartheta_j^k}) \,. \tag{1}$$

Each cortical map is associated with its own proximity measure which best suits to evaluate the similarity between the input and the weight vector, permitting more importance to the underlying input stimuli structure. Even though many self organization processes utilizes Euclidean distance, it does not always work well for multimodal data due to their temporal behavior and for some other complex data structures [11]. We define a unified framework where the proximity between any input and weight vector is estimated by considering the most suitable similarity measure in order to accurately adjust the neuronal responses towards their input stimuli. A prominent characteristic of our neural network model is the ability to deal with dynamic temporal sequences by employing *dynamic time warping*(DTW) technique [12].

After locating the best matching neuron $\omega^k \in C^k$, for a given input stimuli $\overrightarrow{x_i^k}$ the next operation is to update the neighborhood $H^k(\omega^k)$ of the neuron ω^k. The widely used method to model the $H^k(\omega^k)$using Gaussian neighborhood function and neural vectors are updated using,

$$\Delta x_i^k = \mu e^{-\frac{\left\|\overrightarrow{r_\omega^k} - \overrightarrow{r_i^k}\right\|^2}{2\sigma^2}} (\overrightarrow{\vartheta_\omega^k} - \overrightarrow{x_i^k}) \,. \tag{2}$$

The parameter μ denotes the learning rate and σ is the amplitude of the neighborhood affected. Both μ and σ are linearly decreasing functions of discrete time index or training iterations. When the input stimuli structure forms variable length of sequences then the above equation (2) cannot be used as it does not support variable length vectors. We have addressed this issue previously [13] by taking the average of two sequences along the warping path (a path calculated when estimating the distance between two sequences using DTW) when updating the neighbors of the GSOM. The proposed neural network architecture utilize this neighborhood update rule (not discussed in this paper and please refer [13] for more details) when it encounters stimulus which are composed of temporal sequences on its receptive field.

There are several rules that the nervous system employs which affect activity dependent refinement of connections during development [14]. Studies of *development plasticity* shows that the alteration of the patterns of the established connections over the course of development are achieved by induced blocking or altering the activity patterns normally present in the brain [15]. The development plasticity is accomplished by particular governing learning rules. Taking inspiration from above phenomenon we believe that the neighbourhood update rule should be adapted according to the underlying input stimuli structure. Such adaptation facilitates more accurate smoothing of neuronal vectors. We define a common neighbourhood update rule Θ^k which takes the following form,

$$\Delta x_i^k = \Theta^k(\mu, \sigma, \overrightarrow{r_\omega^k}, \overrightarrow{r_i^k}, \overrightarrow{\vartheta_\omega^k}, \overrightarrow{x_i^k}) \,. \tag{3}$$

The characteristics of Θ^k is determined based on the underlying input stimuli structure and most of the cases it takes the form of (2).

Once the winner is identified and its neighbourhood updated, the next step is to establish the passive lateral connections between the two growing self organizing maps. Since the proposed architecture is composed of two cortical areas modelled by GSOM, we set k to be 1 and 2. The learning of the passive lateral connections is carried out during the smoothing phase of the GSOM learning. When a neuron $\omega \in C^k$ wins for an input stimulus $\overrightarrow{x_i^k}$, the frequency vector $\overrightarrow{\tau_\omega^k}$ updates (increment) the frequency or count of input stimulus at index i and increment the neuron's hit counter λ by one.

The passive lateral connections $\pi_t \in \mathbb{R}_0^+$ are learned using the Hebb rule as follows where t denotes the discrete time index or training epoch in the smoothing phase.

$$\pi_t = \pi_{t-1} + \xi\theta^1\theta^2 \,. \tag{4}$$

θ^1 and θ^2 is the neuronal activity levels of the wining neurons of C^1 and C^2 respectively and it is evaluated using a Gaussian transfer function based on similarity between input stimuli and synaptic weight of the wining neuron.

$$y^k = D^k(\overrightarrow{x_i^k}, \overrightarrow{\vartheta_\omega^k}) \,, \tag{5}$$

where $y^k \in [0,1]$.

$$\theta^k(y) = ae^{-}\left(\frac{(y-b)^2}{2c^2}\right). \tag{6}$$

where a, b, c are constants. Once the lateral connections of the wining neurons from two cortical areas are updated the connection strength is normalized such that $\pi_t \in [0,1]$.

The parameter ξ is the learning rate and it is given by the proximity between the probability distribution of the input stimuli of the wining neurons of the two cortical areas. In order to calculate ξ for any given pair of wining neurons the frequency distribution of the wining neurons should be transformed to their probability distribution. This probability distribution characterizes the probability of responding to particular input stimuli. If the frequency count for an input stimulus is 0, we still assign a probability for that input since there is a little chance that the neuron will respond to that stimulus in future. Individual frequency values $f_\varepsilon \in \overrightarrow{\tau_\omega^k}$ of the wining neuron are converted to its corresponding probability values ρ_ε as follows.

$$\rho_\varepsilon = \begin{cases} 1/M\lambda \,, & f_\varepsilon = 0 \\ (f_\varepsilon/\lambda) - \left[\frac{M-\varphi}{\varphi M\lambda}\right], & f_\varepsilon > 0 \end{cases} . \tag{7}$$

where φ is the number of frequency values satisfying, $f_\varepsilon = 0$.

Once the probability distributions were obtained for the wining neurons, we then employed the Kullback–Leibler divergence [16] to measure the similarity between two probability distributions. Let $P_1(\varepsilon)$and $P_2(\varepsilon)$be two discrete probability distribution and the similarity between these two distributions can be estimated by,

$$\mathcal{F}(P_1||P_2) = \sum_i P_1(i) \log\frac{P_1(i)}{P_2(i)} \,. \tag{8}$$

Since $\mathcal{F}(P_1||P_2)$ is not symmetric the distance between two distribution is evaluated by taking the average,

$$\mathcal{G}(P_1, P_2) = \frac{\mathcal{F}(P_1||P_2)+\mathcal{F}(P_2||P_1)}{2} \tag{9}$$

Since $\mathcal{G}(P_1, P_2) \geq 0$ and high dissimilar distributions yields larger real values we restrict the range of the $\mathcal{G}(P_1, P_2)$ and calculate the inverse value using,

$$\xi = \alpha e^{-}\left(\frac{(g-\beta)^2}{2\gamma^2}\right) ; g = \mathcal{G}(P_1, P_2) . \tag{10}$$

The output of the above equation is used in (4) to evaluate the learning rate of the passive lateral connection learning. The main motivation to consider the underlying input stimuli distribution is, the correlation between neurons in two different cortical maps is high if the voronoi regions of these neurons are similar [17]. Even though the similarity between two input stimuli distribution can be estimated using frequency distribution itself, the proposed method follows an additional step to evaluate their probability distribution. This is because probability distribution gives an importance to the input stimuli having no frequency count and Kullback–Leibler divergence is a widely used information theoretic approach to evaluate two given data distribution.

GSOM Learning. As described previously the GSOM network is first initialized, then followed by the growing phase. During the growing phase an event sequence $\vec{E}$ is presented to the two cortical maps and the training process continues iteratively until a predefined convergence criterion is met. We employed a fixed number of iterations as the convergence criteria in both growing and smoothing phases. (The network node growth rate is also a viable option for convergence criteria). The growth threshold GT is calculated as,

$$GT = -\ln(SF) . \tag{11}$$

where SF is the spread factor $(0 \leq SF \leq 1)$ which controls the growth of the network. We present the input $\overrightarrow{x_i^k} \in \mathrm{R}^{N_k}$ to the map C^k to find the best matching neuron using any proximity measure $D^k \in [0,1]$ and update the neighborhood of the wining neuron as described in the above section. The accumulated error value TE_ω of the winner is increased by the difference $D^k(\overrightarrow{x_i^k}, \overrightarrow{\vartheta_\omega^k})$ between the input $\overrightarrow{x_i^k}$ and weight vector of the wining neuron ω.When $TE_\omega \geq GT$ nodes are grown if ω is a boundary node or else the error is distributed to its neighbors (refer to [9] for more details on GSOM learning algorithm). The new neurons' weight vectors are initialized to match the neighboring neurons' weights. At each iteration of the growing phase, all the input stimuli in $\vec{E}$ are presented to both maps and then neurons are grown according to the network error.

The smoothing phase occurs after the growing phase in the learning process establishing the passive lateral connections. Once the node growing phase is complete, the weight adaptation is continued with a lower rate of adaptation. The learning rate (μ) is reduced and the neighborhood is set smaller than the growing phase. During the smoothing phase the input stimuli set $\vec{E}$ is repeatedly presented to both the maps and the lateral connections are updated between wining neurons in the maps and the weights of the neurons are smoothed until convergence.

Parameter Selection. The two growing self-organization maps utilizes the same spread factor (SF) and learning rate (μ). Usually the learning rate and spread factor is set to a smaller value around 0.3. In equation (6) we set a = 1.0, b = 0.0 and c = 0.45 allowing the maximum activation level to be 1.0 and minimum activation level to be 0.0846. Kullback–Leibler divergence yield 0.0 if the two probability distributions are identical. If the two distributions are totally different the Kullback–Leibler divergence yield larger positive real values. In equation (10) we set a larger value for the parameter γ in order to increase the deviation of the Gaussian transfer function. Lower deviation results smaller learning rate even for small K-L distance. Hence the learning of passive lateral connections may be affected. In order to avoid this we set $\gamma = 0.9$. Fig .2 illustrates the effect of the Gaussian deviation parameter γ. The other two parameters α and β are set to 1.0 and 0.0 respectively.

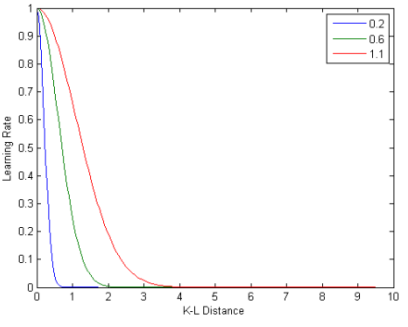


Fig. 2. The effect of Gaussian deviation parameter γon the learning rate. The graph is drawn for three γ levels.

3 Experimental Design and Results

In order to evaluate and validate the performance of the proposed neural network architecture, two sets of experiments were designed using real data sets. The first experiment was drawn from the computer vision where we compare edge maps and color maps corresponding to set of shape images. This experiment primarily focuses on creating two maps for input data pertaining to a single modal. These images were obtained from publically available image database. The original images (464 instances were selected) were manually colored and then features were extracted according to the MPEG-7 Descriptor protocols [18]. The color features were extracted according to the MPEG-7 Scalable Color Descriptors and edge features were extracted according to the MPEG-7 Edge Histogram Descriptors. During the learning of the cortical maps the cosine distance was used for both color and edge maps since it is a widely used proximity measure in computer vision. The neighborhood learning mechanism was the Kohonen rule as specified in (2).

The second experiment was designed to demonstrate the performance under two modalities. Two maps were created for audio and visual stimuli for the data obtained from a lip reading application [19]. The audio features were extracted using in-house

written MATLAB code. Each audio file was segmented by employing a small window and the first 12 mel-frequency cepstrum coefficients (MFCC) were extracted for each segment. The number of sequences or segments depends on the duration of the utterance time and since this time varies from speaker to speaker, the final data set consists of varying length sequences. The dynamic time warping distance was used as the similarity measure for both audio and visual data. The neighbourhood learning rule was based on the concept of average between two sequences (over the warping path) and this technique is described in [13].

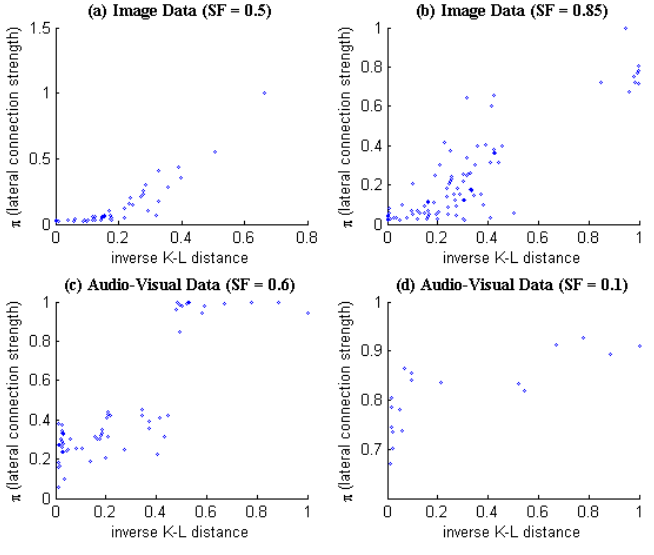


Fig. 3. The relationship between inverse K-L distance for the image and the audio-visual data under different spread factor levels. For the image data threshold is 0.02 and for the audio-visual data the threshold is 0.01.

Several experiments were carried out under different parameter settings for both image and lip reading data sets. During the learning phase the input stimuli (input features) are applied to the receptive fields of the cortical maps and the learning process continues as described in the section 2. Once the cortical maps complete learning for a given input stimuli set, the individual neurons in the maps represent abstract concept related to the corresponding input stimuli features. The passive lateral connections reflect the association between the concepts in the cortical maps. These concepts can be thought of as clusters of the input data set. We evaluate the similarity of input stimuli recognition by the two neurons which are connected by a lateral connection, using inverse K-L distance according to (10). Next, we compare the inverse K-L distance with the passive lateral connection strength (π) between two neurons. A threshold was defined to eliminate the insignificant passive lateral connections .Fig. 3 demonstrates the relationship between inverse K-L distance and π for the experiments done on the data sets described above. For each of the cases (a), (b), (c) and (d) the Pearson's Correlation Coefficient were calculated and it was 0.88,

0.873, 0.847 and 0.753 respectively. The scatter plot results and the Pearson's Correlation Coefficients demonstrate the positive correlation between neurons' association and its lateral connection strength. The scatter plots illustrate higher correlation between inverse K-L and lateral connection strength for image data over audio-visual data but the audio-visual concepts (clusters) demonstrated greater correlation when the underlying class labels were used in comparison.

4 Discussion and Conclusion

An unsupervised laterally connected neural network model is proposed in this paper. The model is composed of two cortical areas modeled by the growing self-organization maps. The proposed model is capable of identifying associations between the two cortical areas. During the learning phase the input features or stimuli arriving at receptive fields are transformed to more abstract concepts, establishing the passive lateral connections which ultimately represent the associations between the concepts learned. The strength of the passive lateral connection reflects frequency of the input stimuli present at the receptive field and the association between the neuronal responses towards to specific stimuli.

Since the passive lateral connection integrates properties for a certain object, our model can also be used to explain the *binding problem* under the theory of feature integration of attention [20]. The learned passive lateral connections facilitate querying for concepts corresponding to another sensory modality (when the stimuli related to this modality is not present in the receptive field) when only one sensory stimuli is available on the receptive field. Also the proposed model can be used in associative pattern discovery in data mining.

The proposed approach highlights the importance of utilizing specialized proximity measures and neighborhood learning rules for complex data structures. Even though the model is described and implemented for two cortical areas, this can be generalized to several cortical areas to integrate stimuli coming from several sensory channels.The most severe drawback of the presented system is so far the cost associated with learning since the current version of the model is implemented in sequential manner. This can be avoided by parallel processing of the input stimuli in the receptive fields. The model is tested upon two types of real data sets and the results exhibit the use of lateral connections in concept association.

References

1. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
2. Torra, V.: On some aggregation operators for numerical information. In: Torra, V. (ed.) Information Fusion in Data Mining. Studies in Fuzziness and Soft Computing, vol. 123, Springer, Heidelberg (2003)
3. Lanckriet, G.R.G., Deng, M., Cristianini, N., Jordan, M.I., Noble, W.S.: Kernel-based data fusion and its application to protein function prediction in yeast. In: Pacific Symposium on Biocomputing 2004, pp. 300–311 (2004)

4. Coen, M.H.: Cross-Modal Clustering. In: Twentieth National Conference on Artificial Intelligence (2005)
5. Wermter, C.W.A.S.: Object Localisation Using Laterally Connected "What" and "Where" Associator Networks. In: International Conference on Artificial Neural Networks (2003)
6. Isokawa, T., Nishimura, H., Kamiura, N., Matsui, N.: Perceptual Binding by Coupled Oscillatory Neural Network. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005, Part I. LNCS, vol. 3696, pp. 139–144. Springer, Heidelberg (2005)
7. Magossoa, E., Cuppini, C., Serino, A., Pellegrino, G.D., Ursino, M.: A Theoretical Study of Multisensory Integration in the Superior Colliculus by a Neural Network Model. Neural Networks 21, 817–829
8. Anastasio, T.J., Patton, P.E.: A Two-Stage Unsupervised Learning Algorithm Reproduces Multisensory Enhancement in a Neural Network Model of the Corticotectal System. The Journal of Neuroscience 23(14) (2003)
9. Alahakoon, D., Halgamuge, S.K., Srinivasan, B.: Dynamic Self-Organising Maps with Controlled Growth for Knowledge Discovery. IEEE Transactions on Neural Networks 11, 601–614 (2000)
10. Hebb, D.O.: The Organization of Behavior. Wiley, New York (1949)
11. Niennattrakul, V., Ratanamahatana, C.A.: On Clustering Multimedia Time Series Data Using K-Means and Dynamic Time Warping. In: International Conference on Multimedia and Ubiquitous Engineering, MUE 2007 (2007)
12. Sankoff, D., Kruskal, J.: Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. CSLI Publications (1999)
13. Fonseka, A., Alahakoon, D., Bedingfield, S.: GSOM Sequence: An Unsupervised Dynamic Approach for Knowledge Discovery in Temporal Data. Paper presented at the IEEE Symposium on Computational Intelligence and Data Mining, Paris, France
14. Butts, D.A.: Retinal Waves: Implications for Synaptic Learning Rules during Development. Neuroscientist (2002)
15. Katz, L.C., Shatz, C.J.: Synaptic activity and the construction of cortical circuits. Science 274, 1133–1138 (1996)
16. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Statistics 22(1), 79–86 (1951)
17. Baruque, B., Corchado, E.: Fusion Methods for Unsupervised Learning Ensembles. Springer, Heidelberg (2010)
18. Sikora, T.: The MPEG-7 Visual Standard for Content Description—An Overview. IEEE Transactions on Circuits and Systems for Video Technology 11(6) (2011)
19. Movellan, J.: Visual Speech Recognition With Stochastic Networks. In: Advances in Neural Information Processing Systems, pp. 851–858 (1995)
20. Treisman, A.M., Gelade, G.: A feature-integration theory of attention. Cognitive Psychology 12(1), 97–136 (1980)

Author Index

Aboutajdine, Driss II-300
Agrawal, Ritesh III-148
Ai, Lifeng II-258
Aihara, Kazuyuki III-381
Alahakoon, Damminda II-193, II-406, II-761
Al-Nuaimi, A.Y.H. I-550
Amin, Md. Faijul I-541, I-550
Amin, Muhammad Ilias I-550
Amrouche, Abderrahmane II-284, II-292
An, Senjian II-681
Ando, Ruo II-28
André-Obrecht, Régine II-300
Aonishi, Toru III-240
Arie, Hiroaki I-501, III-323
Asbai, Nassim II-284
Awano, Hiromitsu III-323

Ban, Sang-Woo III-557
Ban, Tao II-18, II-45
Barczak, Andre L. III-495
Barros, Allan Kardec I-54, II-545
Barua, Sukarna II-735
Basirat, Amir H. II-391
Bdiri, Taoufik II-71
Belatreche, A. I-461
Belguith, Lamia Hadrich III-131
Bennani, Younès I-570, II-745
Bhowmik, Tapan Kumar III-538
Bian, Wei I-589, III-657
Bosse, Tibor I-423
Both, Fiemke III-700
Bouguila, Nizar II-71, II-125, II-276, III-514
Bu, Jiajun I-589, III-86, III-667

Cabanes, Guénaël II-745
Cao, Jianting I-279, I-314
Cao, Xiao-Zhong III-711
Caselles, Vicent II-268
Catchpoole, Daniel I-113
Câteau, Hideyuki III-485
Cavalcante, André I-54
Ceberio, Josu II-461
Cerezuela-Escudero, E. I-190
Chai, Guangren III-113
Chan, Jonathan H. I-676
Chandra, Vikas I-423
Chandran, Vinod III-431
Chang, Jyh-Yeong III-356
Chang, Qingqing I-241, I-257
Chen, Chuantong II-37
Chen, Chun I-589, III-86, III-667
Chen, Jing II-89
Chen, Jin-Yuan II-169
Chen, Li-Fen I-684
Chen, Qingcai II-211, III-158
Chen, Shi-An I-701, I-717
Chen, Xiaoming II-109, II-681
Chen, Xilin I-172
Chen, Yanjun III-676
Chen, Yin-Ju II-63, II-185
Chen, Yong-Sheng I-684
Chen, Yuehui I-107
Cheng, Shi II-228
Cheng, Yu I-233
Chetty, Girija III-1
Chetty, Madhu I-97, I-625, I-636, II-248, III-36, III-719
Cheu, Eng Yeow I-493
Chiu, Chien-Yuan III-292
Cho, Kenta II-358
Cho, Minkook II-350
Cho, Sung-Bae I-38, III-57
Choi, Seungjin II-325
Choi, Sung-Do I-217
Chung, Younjin II-133
Cichocki, Andrzej I-279, I-287, I-322, II-663
Constantinides, A.G. III-373
Coppel, Ross I-97, III-719
Cornuéjols, Antoine I-580
Cui, Yuwei III-210

Daoudi, Khalid II-125, II-300
Davies, Sergio III-259
Debyeche, Mohamed II-284, II-292

Deng, Jeremiah III-449
Dhoble, Kshitij I-451, III-230
Diaz-del-Rio, F. I-199
Ding, Yuxin II-374, III-113, III-315
Domínguez-Morales, M. I-190, I-199
Dong, Li III-315
Dou, Zhang I-46
Duan, Lijuan I-182, I-296
Duch, Włodzisław II-726

Eto, Masashi II-18

Fahey, Daniel II-143
Fan, Ke II-109, II-681
Fan, Shixi III-121
Fan, Wentao II-276
Fang, Fang I-172
Fangyu, He I-265
Farkaš, Igor I-443
Feigin, Valery I-129
Fernández-Redondo, Mercedes II-572, II-580, II-588
Fidge, Colin II-258
Fiori, Simone III-365
Fonseka, Asanka II-761
Fresneau, Dominique II-745
Fu, Zhouyu II-490
Fujita, Kazuhisa III-251
Fukushima, Kunihiko II-628
Funase, Arao I-322
Furber, Steve III-424
Furukawa, Tetsuo II-618

Galluppi, Francesco III-424
Ganegedara, Hiran II-193
Gao, Pengyuan II-554
Gao, Su-Yan II-53
Gao, Wen I-172
Ge, Rendong II-554
Ge, Shuzhi Sam I-225
Gedeon, Tom I-396, II-143, III-348
Gerla, Václav I-388
Gerritsen, Charlotte III-26
Ghoshal, Ranjit III-538
Gleeson, Andrew I-113
Go, Jun-Ho II-316
Gönen, Mehmet II-500
Gong, Bin II-37, II-45
Gong, Dunwei II-445
Gori, Marco I-28, II-519
Graja, Marwa III-131
Grozavu, Nistor I-570
Gu, Jili I-182
Gu, Jing-Nan I-380
Guo, Ping II-203, III-459, III-467
Guo, Shanqing II-18, II-37, II-45
Gustafsson, Lennart I-413

Hafiz, Abdul Rahman I-541
Hamed, Haza Nuzly Abdull II-160
Hammer, Barbara II-481
Han, Jiqing II-646
Han, Qi I-164
Hao, Hong-Wei III-711
Harada, Hidetaka III-332
Hasegawa, Osamu III-47
Hashemi, Mitra II-220
Hayakawa, Yoshihiro III-389
Hayashi, Hatsuo I-370
Hayashi, Isao II-628
He, Hongsheng I-225
He, Tian-Tian III-711
He, Xiangjian III-547, III-756
He, Xin I-164
Helbach, Jonathan III-639
Hernández-Espinosa, Carlos II-572, II-580, II-588
Hibi, Ryota II-655
Hirasawa, Hajime III-684
Hirose, Akira I-526
Ho, Kevin III-268
Ho, Nicholas I-113
Ho, Shiu-Hwei II-63, II-185
Hoang, Tuan I-692
Honkela, Timo III-167
Hoogendoorn, Mark III-700
Hori, Gen I-314
Hossain, Emdad III-1
Hu, Bin II-89
Hu, Fanxing III-187
Hu, Jinglu I-233
Hu, Jun I-493
Hu, Rukun III-467
Hu, Yingjie I-129, I-646
Huang, Gang I-241, I-257
Huang, Kaiqi III-629
Huang, Kaizhu II-151, III-747
Huang, Kejie I-469
Huang, Mao Lin I-113, II-99
Huang, Pin II-406

Huang, Tze-Haw II-99
Huang, Weicheng III-9
Huang, Weiwei I-477, I-485
Huayaney, Frank L. Maldonado III-381
Hwang, Byunghun II-342

Iida, Munenori III-240
Iima, Hitoshi I-560
Ikeda, Kazushi I-532, II-358
Ikegaya, Yuji I-370
Ikenoue, Tsuyomu II-117
Ikeuchi, Ryota I-532
Inoue, Daisuke II-18
Ishak, Dahaman III-730
Ishikawa, Takumi III-95
Islam, Md. Kamrul I-625, I-636
Islam, Md. Monirul II-735
Ito, Ryo II-606
Ito, Yoshifusa II-596
Izumi, Hiroyuki II-596

Jaber, Ghazal I-580
Jamdagni, Aruna III-756
Jang, Young-Min I-138
Jankowski, Norbert II-238
Jaoua, Maher III-131
Jeong, Sungmoon I-501
Jia, Wenjing III-547
Jiang, Xiaomin II-117
Jiang, Yong II-169, II-177, II-399
Jimenez-Fernandez, A. I-190
Jimenez-Moreno, G. I-190, I-199
Jin, Jesse S. II-99
Jin, Jing I-273, I-287
Jing, Huiyun I-164
Antoine, Cornuéjols I-608
Jourani, Reda II-300
Jung, Ilkyun III-557

Kamiji, Nilton Liuji III-684
Kandemir, Melih II-500
Kanehira, Ryu III-76
Kasabov, Nikola I-129, I-451, I-646, II-160, II-718, III-230
Kashimori, Yoshiki I-62
Kaski, Samuel II-500
Kawasue, Kikuhito III-573
Kawewong, Aram III-47
Khan, Asad I. II-391
Khor, Swee Eng I-668

Kil, Rhee Man III-774
Kim, Bumhwi III-416
Kim, Cheol-Su II-342
Kim, Ho-Gyeong III-774
Kim, JungHoe I-306
Kim, Min-Young II-342
Kim, Seongtaek II-316
King, Irwin III-148, III-747
Kitahara, Michimasa I-509
Kivimäki, Ilkka III-167
Ko, Li-Wei I-717
Kobayashi, Kunikazu III-76
Kobayashi, Masaki I-509
Koike, Yuji III-47
Kortkamp, Marco III-639
Koya, Hideaki III-621
Kugler, Mauricio II-545
Kühnel, Sina III-639
Kurashige, Hiroki III-485
Kuremoto, Takashi III-76
Kuriya, Yasutaka III-611
Kuroe, Yasuaki I-560
Kurogi, Shuichi I-70, III-9, III-621
Kurokawa, Makoto III-684
Kwak, Ho-Wan I-138

Laaksonen, Jorma III-737
Labiod, Lazhar II-700, II-709
Lai, Jianhuang II-109
Lam, Ping-Man III-373
Lam, Yau-King I-654, I-662
Lang, Bo III-601
Le, Trung I-692, II-529, II-537
Lee, Giyoung III-557
Lee, Jong-Hwan I-306
Lee, Minho I-138, I-501, II-342, III-340, III-416, III-557
Lee, Sangil I-138
Lee, Seung-Hyun III-57
Lee, Soo-Young I-217, III-774
Lee, Wee Lih I-352
Lee, Wono III-557
Lee, Young-Seol I-38
Lee, Yun-Jung II-342
Lester, David III-259
Leung, Chi-Sing I-654, I-662, III-268
Leung, Chi-sing III-276, III-373
Leung, Yee Hong I-352
Léveillé, Jasmin II-628

Lhotská, Lenka I-388, I-443
Li, Bao-Ming III-187
Li, Dong-Lin III-356
Li, Fang II-416
Li, JianWu III-649
Li, Jianwu II-382
Li, Jinlong II-453
Li, Jun I-329
Li, Liangxiong II-45
Li, Ming II-399
Li, Qing III-711
Li, Tao III-573
Li, Xiaolin III-307
Li, Xiaosong II-11
Li, Yang I-725
Li, Yuanqing III-210
Li, Yun II-53
Liang, Haichao III-522
Liang, Wen I-129
Liang, Xun II-510
Liao, Shu-Hsien II-63, II-185
Lim, Chee Peng I-668
Lim, CheePeng III-730
Lim, Suryani III-36
Lin, Chin-Teng I-701, I-717, III-356
Lin, Fengbo II-37
Lin, Jia-Ping I-684
Lin, Lei I-121
Lin, Lili III-592
Linares-Barranco, A. I-190, I-199
Liu, Bing II-638
Liu, Bingquan II-671
Liu, Cheng-Lin II-151, III-747
Liu, Hong-Jun I-380
Liu, Lijun II-554
Liu, Lixiong III-459
Liu, Ming III-307
Liu, Ren Ping III-756
Liu, Ruochen II-435
Liu, Wanquan II-109, II-681
Liu, Xiao I-589, III-657
Liu, Yunqiang II-268
Lopes, Noel II-690, III-766
López-Torres, M.R. I-190, I-199
Lozano, Jose A. II-461
Lu, Bao-Liang I-380, I-709, I-725, I-734
Lu, Guanzhou II-453
Lu, Guojun II-490
Lu, Hong-Tao I-380, I-404
Lu, Yao III-649
Lu, Zhanjun III-315
Lucena, Fausto II-545

Ma, Peijun III-18
Ma, Wanli I-692, II-529, II-537
Ma, Yajuan II-435
Maguire, L.P. I-461
Mallipeddi, Rammohan I-138
Martin, Christine I-608
Maruno, Yuki II-358
Mashrgy, Mohamed Al II-125
Matharage, Sumith II-406
Matsubara, Takashi III-395
Matsuda, Yoshitatsu I-20
Matsuo, Takayuki III-381
Matsuzaki, Shuichi I-360
McGinnity, T.M. I-461
Meechai, Asawin I-676
Melacci, Stefano I-28, II-519
Mendiburu, Alexander II-461
Meng, Xuejun III-113
Meybodi, Mohammad Reza II-220
Miao, Jun I-172, I-182, I-296
Miki, Tsutomu III-332
Milone, Mario III-103
Mineishi, Shota I-70
Mitleton-Kelly, Eve I-423
Mitsukura, Yasue I-46
Miwa, Shinsuke II-28
Mohammed, Rafiq A. II-1
Mohemmed, Ammar II-718, III-230
Moore, Philip II-89
Morgado, A. I-190
Morie, Takashi III-381, III-522
Morshed, Nizamul II-248
Mount, William M. I-413
Murase, Kazuyuki I-541, I-550, II-735
Murshed, Manzur I-636
Mutoh, Yoshitaka I-62

Nadif, Mohamed I-599, II-700, II-709
Nagatomo, Satoshi III-573
Nagi, Tomokazu III-621
Nakajima, Koji III-389
Nakao, Koji II-18
Nakayama, Yuta II-606
Nanda, Priyadarsi III-756
Negishi, Yuna I-46
Nguyen, Phuoc II-537
Nguyen, Quang Vinh I-113

Nie, Dan I-734
Ning, Ning I-469
Nishida, Takeshi I-70, III-9, III-621
Nishide, Shun III-323
Nishio, Kimihiro III-506
Nitta, Tohru I-519
Niu, Xiamu I-164
Noh, Yunseok II-316
Nuntalid, Nuttapod I-451, III-230

Obayashi, Masanao III-76
Ogata, Tetsuya III-323
Ogawa, Takashi II-612
Oh, Myungwoo II-366
Ohnishi, Noboru I-54, II-545
Oja, Erkki III-167, III-737
Okada, Masato III-240
Okamoto, Yuzo II-358
Okuno, Hiroshi G. III-323
Okuno, Hirotsugu III-416
Omori, Toshiaki III-240
Onishi, Akinari I-279, I-287

Pang, Paul II-1
Papliński, Andrew P. I-413
Park, Hyeyoung II-335, II-350
Park, Hyung-Min II-342, II-366
Park, Seong-Bae II-316
Park, Yunjung I-501
Parui, Swapan K. III-538
Pathak, Vinay K. II-753
Paukkeri, Mari-Sanna III-167
Paz, R. I-190
Peng, Fei II-425
Plana, Luis A. III-424
Prom-on, Santitham I-676

Qi, Sihui I-155
Qiao, Haitao I-182
Qiao, Yu I-241, I-249, I-257
Qin, Quande II-228
Qin, Zengchang III-139
Qing, Laiyun I-172, I-182
Qiu, Wei I-79
Qiu, Zhi-Jun I-337

Rajapakse, Jayantha II-406, II-761
Ren, Yuan III-582
Reyes, Napoleon H. III-495
Ribeiro, Bernardete II-690, III-766
Rogovschi, Nicoleta I-599
Roy, Anandarup III-538

Saifullah, Mohammad I-88
Saito, Toshimichi II-606, II-612
Sakaguchi, Yutaka III-95
Sakai, Ko I-79
Sameshima, Hiroshi II-117
Samura, Toshikazu I-370
Sasahara, Kazuki III-66
Sato, Fumio III-47
Sato, Shigeo III-389
Sato, Yasuomi D. I-370, III-611
Satoshi, Naoi III-747
Schack, Thomas III-639
Schleif, Frank-Michael II-481
Schliebs, Stefan II-160, II-718
Schrauwen, Benjamin III-441
Schütz, Christoph III-639
Seera, Manjeevan III-730
Sefidpour, Ali III-514
Seo, Jeongin II-335
Setoguchi, Hisao II-358
Shah, Munir III-449
Shang, Ronghua II-435
Sharma, Dharmendra I-692, II-529, II-537
Sharma, Nandita III-348
Sharp, Thomas III-424
Shen, Chengyao I-225
Shi, Li-Chen I-709, I-725
Shi, Luping I-469
Shi, Yuhui II-228
Shi, Ziqiang II-646
Shibata, Katsunari III-66
Shiina, Yusuke I-360
Shin, Heesang III-495
Sima, Haifeng III-459
Simoff, Simeon I-113
Situ, Wuchao I-662
Sjöberg, Mats III-737
Sokolovska, Nataliya III-103
Son, Jeong-Woo II-316
Song, Anjun I-155
Song, Hyun Ah I-217
Song, Mingli I-589, III-86, III-657, III-667
Song, Sanming I-1, I-435
Sootanan, Pitak I-676
Sota, Takahiro III-389

Srinivasan, Cidambi II-596
Srivastava, Vivek II-753
Steinhöfel, K. I-625
Stockwell, David III-530
Su, Xiaohong III-18
Sum, John Pui-Fai III-268, III-276, III-373
Sun, Chengjie I-121, II-671
Sun, Jing II-445
Sun, Jun III-747
Sun, Rui-Hua I-725
Sun, Xiaoyan II-445
Suzuki, Yozo I-509
Szymański, Julian II-726

Takahashi, Norikazu II-655
Takahashi, Toru III-323
Takatsuka, Masahiro II-133
Takeuchi, Yoshinori I-54
Takumi, Ichi I-322
Tamura, Hiroki II-117
Tan, Chin Hiong I-485, I-493
Tan, Phit Ling I-668
Tan, Qianrong III-299
Tan, Shing Chiang I-668
Tan, Tele I-352
Tan, Zhiyuan III-756
Tanaka, Hideki III-381
Tang, Buzhou II-211, III-177
Tang, Huajin I-477, I-485, I-493
Tang, Huixuan III-582
Tang, Ke II-425
Tang, Long III-36
Tang, Maolin II-258
Tang, Yan III-113
Tani, Jun I-501, III-323
Tanigawa, Shinpei I-560
Tanno, Koichi II-117
Tao, Dacheng I-329, I-589, III-86, III-629, III-657, III-667
Tarroux, Philippe I-580
Teytaud, Olivier III-103
Tian, Xin I-337
Ting, Kai-Ming II-490
Tirunagari, Santosh III-167
Tjondronegoro, Dian III-431
Torikai, Hiroyuki III-395, III-405
Torres-Sospedra, Joaquín II-572, II-580, II-588
Tran, Dat I-692, II-529, II-537
Treur, Jan I-9, III-197, III-217
Tripathi, Bipin K. II-753
Tsang, P.W.M. I-654, I-662
Tscherepanow, Marko II-562, III-639
Tsukazaki, Tomohiro I-70

Ullah, A. Dayem I-625
Umair, Muhammad III-217
Usowicz, Krzysztof II-238
Usui, Shiro III-684

Vavrečka, Michal I-388, I-443
Verma, Brijesh III-292, III-530
Vinh, Nguyen Xuan I-97, II-248, III-719
Vo, Tan I-396

Wada, Yasuhiro I-360
Waegeman, Tim III-441
Wal, C. Natalie van der I-423
Wan, Tao III-139
Wang, Bin III-475
Wang, Dandan II-211, III-158
Wang, Fengyu II-37, II-45
Wang, J. I-461
Wang, Kuanquan III-676
Wang, Lan I-233
Wang, Lin I-617
Wang, Qiang I-241, I-257
Wang, Senlin III-667
Wang, Sheng III-547
Wang, Xiaolong I-121, II-211, II-671, III-121, III-158, III-177
Wang, Xiao-Wei I-734
Wang, Xin III-475
Wang, Xingyu I-273, I-287
Wang, Xuan I-121, III-121, III-177
Wang, Xuebin I-296
Wang, Yuekai I-209
Wang, Yu-Kai I-701
Wangikar, Pramod P. I-97, III-719
Webb, Andrew III-259
Wei, Chun-Shu I-717
Wei, Hui III-187, III-582, III-601
Weng, Jiacai III-158
Weng, Juyang I-209
Wenlu, Yang I-265
Woodford, Brendon III-449
Wu, Chunpeng I-182, I-296
Wu, Jie I-709
Wu, Qiang II-663, III-547

Wu, Qiufeng III-676
Wu, Si III-210
Wu, Weigen III-299
Wu, Xiaofeng I-209
Wu, Yang I-146

Xia, Shu-Tao II-169, II-177, II-399
Xiao, Min II-374
Xiao, Yi I-662
Xie, Bo III-629
Xinyun, Chen I-265
Xu, Bingxin II-203
Xu, Bo III-747
Xu, Jianhua II-79
Xu, Jingru I-107
Xu, Kunhan I-345
Xu, Li III-18
Xu, Ruifeng III-177
Xu, Shanshan III-284
Xu, Sheng III-284
Xu, Xi III-711
Xudong, Huang I-265

Yagi, Tetsuya III-416
Yakushiji, Sho II-618
Yamada, Masahiro III-684
Yamaguchi, Kazunori I-20
Yamamoto, Kazunori III-684
Yamashita, Yukihiko II-471
Yamashita, Yutaro III-405
Yan, Zehua II-416
Yan, Ziye III-649
Yang, Fan I-617
Yang, Jie I-241, I-249, I-257, III-547
Yang, Li II-117
Yang, Peipei II-151
Yang, Xiaohong III-121
Yang, Yang I-155
Yang, Zhen I-182, I-296
Yao, Hongxun I-1, I-435
Yao, Xin II-453
Yasuda, Taiki III-506
Ye, Ning III-284
Yessad, Dalila II-292
Yi, Kaijun I-469
Yi, Si-Hyuk III-57
Yin, Xing III-299
Yin, Xu-Cheng III-711
Yokota, Tatsuya II-471
Yoshida, Shotaro I-526
Yoshinaga, Shoichi III-621
You, Qingzhen II-374
Yu, Hongbin I-404
Yu, Jiali I-485
Yu, Qiang I-493
Yu, Xiaofeng III-148
Yu, Yang I-121
Yun, Jeong-Min II-325
Yuno, Hiroshi III-9

Zajac, Remi III-148
Zhang, Danke III-210
Zhang, Dengsheng II-490
Zhang, Deyuan II-671
Zhang, He III-737
Zhang, Hong-Yan I-337
Zhang, Hua III-592
Zhang, Li II-638
Zhang, Ligang III-431
Zhang, Liming III-475
Zhang, Liqing I-146, I-279, II-308, II-663, III-565
Zhang, Luming I-589, III-86, III-657, III-667
Zhang, Qi I-296
Zhang, Qing II-382, III-340
Zhang, Tao I-345
Zhang, Xiaowei II-89
Zhang, Yaoyun III-177
Zhang, Yu I-273, I-279, I-287
Zhang, Zhiping II-382
Zhao, Bin III-315
Zhao, Haohua II-308
Zhao, Qi III-139
Zhao, Qibin I-279, I-287
Zhao, Xing II-177
Zheng, Hai-Tao II-169, II-177, II-399
Zheng, Shuai III-629
Zheng, Tieran II-646
Zhou, Bolei III-692
Zhou, Di II-374
Zhou, Guoxu I-287
Zhou, Haiyan I-296
Zhou, Lin II-89
Zhou, Liuliu III-284
Zhou, Rong III-565
Zhou, Wei-Da II-638
Zhou, Wenhui III-592
Zhou, Zhuoli III-86
Zhu, Dingyun II-143

Zhu, Fa III-284
Zhu, Mengyuan III-692
Zhu, Xibin II-481
Zhu, Yuyuan I-241, I-257
Ziou, Djemel II-276
Zuo, Qingsong III-601
Zuo, Wangmeng III-676