De-Shuang Huang
Phalguni Gupta
Ling Wang
Michael Gromiha (Eds.)

Communications in Computer and Information Science 375

Emerging Intelligent Computing Technology and Applications

9th International Conference, ICIC 2013
Nanning, China, July 2013
Proceedings

Springer

Communications in Computer and Information Science 375

Editorial Board

De-Shuang Huang Phalguni Gupta
Ling Wang Michael Gromiha (Eds.)

Emerging Intelligent Computing Technology and Applications

9th International Conference, ICIC 2013
Nanning, China, July 28-31, 2013
Proceedings

Springer

Volume Editors

De-Shuang Huang
Tongji University, Shanghai, China
E-mail: dshuang@tongji.edu.cn

Phalguni Gupta
Indian Institute of Technology Kanpur, India
E-mail: pg@cse.iitk.ac.in

Ling Wang
Tsinghua University, Beijing, China
E-mail: wangling@tsinghua.edu.cn

Michael Gromiha
Indian Institute of Technology (IIT) Madras
Chennai, India
Email: gromiha@iitm.ac.in

ISSN 1865-0929 e-ISSN 1865-0937
ISBN 978-3-642-39677-9 e-ISBN 978-3-642-39678-6
DOI 10.1007/978-3-642-39678-6
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013942872

CR Subject Classification (1998): F.1.1, I.2.10, I.4.7, I.5, I.6.4, I.6.6, H.2.8

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

The International Conference on Intelligent Computing (ICIC) was started to provide an annual forum dedicated to the emerging and challenging topics in artificial intelligence, machine learning, pattern recognition, image processing, bioinformatics, and computational biology. It aims to bring together researchers and practitioners from both academia and industry to share ideas, problems, and solutions related to the multifaceted aspects of intelligent computing.

ICIC 2013, held in Nanning, China, July 28–31, 2013, constituted the 9th International Conference on Intelligent Computing. It built upon the success of ICIC 2012, ICIC 2011, ICIC 2010, ICIC 2009, ICIC 2008, ICIC 2007, ICIC 2006, and ICIC 2005 that were held in Huangshan, Zhengzhou, Changsha, China, Ulsan, Korea, Shanghai, Qingdao, Kunming, and Hefei, China, respectively.

This year, the conference concentrated mainly on the theories and methodologies as well as the emerging applications of intelligent computing. Its aim was to unify the picture of contemporary intelligent computing techniques as an integral concept that highlights the trends in advanced computational intelligence and bridges theoretical research with applications. Therefore, the theme for this conference was "Emerging Intelligent Computing Technology and Applications". Papers focusing on this theme were solicited, addressing theories, methodologies, and applications in science and technology.

ICIC 2013 received 561 submissions from 27 countries and regions. All papers went through a rigorous peer-review procedure and each paper received at least three review reports. Based on the review reports, the Program Committee finally selected 192 high-quality papers for presentation at ICIC 2013, included in three volumes of proceedings published by Springer: one volume of *Lecture Notes in Computer Science* (LNCS), one volume of *Lecture Notes in Artificial Intelligence* (LNAI), and one volume of *Communications in Computer and Information Science* (CCIS).

This volume of *Communications in Computer and Information Science* (CCIS) includes 39 papers.

The organizers of ICIC 2013, including Tongji University and Guangxi University for Nationalities, made an enormous effort to ensure the success of the conference. We hereby would like to thank the members of the Program Committee and the referees for their collective effort in reviewing and soliciting the papers. We would like to thank Alfred Hofmann from Springer for his frank and helpful advice and guidance throughout and for his continuous support in publishing the proceedings. In particular, we would like to thank all the authors for contributing their papers. Without the high-quality submissions

from the authors, the success of the conference would not have been possible. Finally, we are especially grateful to the IEEE Computational Intelligence Society, the International Neural Network Society, and the National Science Foundation of China for their sponsorship.

May 2013

De-Shuang Huang
Phalguni Gupta
Ling Wang
Michael Gromiha

ICIC 2013 Organization

General Co-chairs

De-Shuang Huang, China
Marios Polycarpou, Cyprus
Jin-Zhao Wu, China

Program Committee Co-chairs

Kang-Hyun Jo, Korea
Pei-Chann Chang, Taiwan, China

Organizing Committee Co-chairs

Yong-Quan Zhou, China
Bing Wang, China

Award Committee Co-chairs

Laurent Heutte, France
Phalguni Gupta, India

Publication Chair

Juan Carlos Figueroa, Colombia

Workshop/Special Session Chair

Vitoantonio Bevilacqua, Italy

Special Issue Chair

Michael Gromiha, India

Tutorial Chair

Luonan Chen, Japan

International Liaison Chair

Prashan Premaratne, Australia

Publicity Co-chairs

Kyungsook Han, Korea
Lei Zhang, China
Ling Wang, China
Valeriya Gribova, Russia

Exhibition Chair

Xing-Ming Zhao, China

Organizing Committee Members

Conference Secretary

Yong Huang, China
Yong Wang, China
Yuanbin Mo, China
Su-Ping Deng, China

Program Committee Members

Andrea Francesco Abate, Italy
Vasily Aristarkhov, Russian Federation
Costin Badica, Romania
Soumya Banerjee, India
Waqas Haider Khan Bangyal, Pakistan
Vitoantonio Bevilacqua, Italy
Shuhui Bi, China
Zhiming Cai, Macau
Chin-Chih Chang, Taiwan, China
Pei-Chann Chang, Taiwan, China
Guanling Chen, USA
Luonan Chen, Japan
Jingdong Chen, China
Songcan Chen, China
Weidong Chen, China
Xiyuan Chen, China
Yang Chen, China
Michal Choras, Poland
Angelo Ciaramella, Italy
Jose Alfredo F. Costa, Brazil
Mingcong Deng, Japan
Eng. Salvatore Distefano, Italy

Mariagrazia Dotoli, Italy
Haibin Duan, China
Hazem Elbakry, Egypt
Karim Faez, Iran
Jianbo Fan, China
Jianwen Fang, USA
Minrui Fei, China
Juan Carlos Figueroa, Colombia
Wai-Keung Fung, Canada
Jun-Ying Gan, China
Liang Gao, China
Xiao-Zhi Gao, Finland
Dunwei Gong, China
Valeriya Gribova, Russia
M. Michael Gromiha, India
Xingsheng Gu, China
Kayhan Gulez, Turkey
Phalguni Gupta, India
Fei Han, China
Kyungsook Han, Korea
Yong-Tao Hao, China
Jim Harkin, UK
Haibo He, USA
Jing Selena He, USA
Laurent Heutte, France
Wei-Chiang Hong, Taiwan, China
Yuexian Hou, China
Heyan Huang, China
Kun Huang, USA
Zhenkun Huang, China
Peter Hung, Ireland
Chuleerat Jaruskulchai, Thailand
Umarani Jayaraman, India
Li Jia, China
Zhenran Jiang, China
Kang-Hyun Jo, Korea
Dong-Joong Kang, Korea
Sanggil Kang, Korea
Muhammad Khurram Khan, Saudi Arabia
Donald H. Kraft, USA
Harshit Kumar, Korea
Yoshinori Kuno, Japan
Takashi Kuremoto, Japan
Vincent C S Lee, Australia
Bo Li, China
Guo-Zheng Li, China
Kang Li, UK
Min Li, China
Shi-Hua Li, China
Xiaoou Li, Mexico
Honghuang Lin, USA
Chunmei Liu, USA
Ju Liu, China
Ke Lv, China
Jinwen Ma, China
Lorenzo Magnani, Italy
Xiandong Meng, USA
Tarik Veli Mumcu, Turkey
Roman Neruda, Czech Republic
Ken Nguyen, USA
Ben Niu, China
Yusuke Nojima, Japan
Sim-Heng Ong, Singapore
Francesco Pappalardo, Italy
Young B. Park, Korea
Surya Prakash, India
Prashan Premaratne, Australia
Seeja K.R., India
Ajita Rattani, Italy
Ivan Vladimir Meza Ruiz, Mexico
Angel D. Sappa, Spain
Li Shang, China
Fanhuai Shi, China
Jiatao Song, China
Stefano Squartini, Italy
Zhan-Li Sun, China
Evi Syukur, Australia
Naoyuki Tsuruta, Japan
Antonio E. Uva, Italy
Katya Rodriguez Vazquez, Mexico
Jun Wan, USA
Bing Wang, China
Lei Wang, China
Ling Wang, China
Shitong Wang, China
Wei Wang, China
Yijie Wang, China
Wei Wei, China
Zhi Wei, China

Qiong Wu, China
Xiaojun Wu, China
Yan Wu, China
Junfeng Xia, China
Shunren Xia, China
Yuanqing Xia, China
Liangjun Xie, USA
Bingji Xu, China
Hua Xu, USA
Shao Xu, Singapore
Zhenyu Xuan, USA
Tao Ye, China
Wen Yu, Mexico
Boyun Zhang, China
Lei Zhang, HongKong, China
Xiang Zhang, USA
Yi Zhang, China
Hongyong Zhao, China
Xing-Ming Zhao, China
Zhongming Zhao, USA
Bo-Jin Zheng, China
Chun-Hou Zheng, China
Fengfeng Zhou, China
Shuigeng Zhou, China
Li Zhuo, China

Additional Reviewers

Marjan Abdechiri
Aliahmed Adam
Erum Afzal
Sabooh Ajaz
Felix Albu
Muhammad Amjad
Deepa Anand
Mary Thangakani Anthony
Vasily Aristarkhov
Sepehr Attarchi
Amelia Badica
Leemon Baird
Abdullah Bal
Waqas Bangyal
Donato Barone
Ye Bei
Olivier Berder
Simon Bernard
Vitoantonio Bevilacqua
Shuhui Bi
Jun Bo
Nora Boumella
Marius Brezovan
Fabio Bruno
Fanliang Bu
Ni Bu
Guorong Cai
Qiao Cai
Francesco Camastra
Giuseppe Carbone
Raffaele Carli
Jair Cervantes
Hyunuk Chae
Aravindan Chandrabose
James Chang
Yuchou Chang
Chun Chen
David Chen
Diyi Chen
Gang Chen
Jianhung Chen
Songcan Chen
Chi-Tai Cheng
Cong Cheng
Ferdinando Chiacchio
Cheng-Hsiung Chiang
Shen Chong
Angelo Ciaramella
Azis Ciayadi
Rudy Ciayadi
Mike Collins
Danilo Comminiello
Carlos Cubaque
Yan Cui
Dajundu
Francesca De Crescenzio
Kaushik Deb
Sara Dellantonio
Jing Deng
Lei Deng
Suping Deng
Somnath Dey
Liya Ding
Sheng Ding
Shihong Ding
Xiang Ding
Joaquín Dopazo
Vlad Dovgalecs
Vladislavs Dovgalecs
Guangyue Du
Ji-Xiang Du
Haibin Duan
Qiqi Duan
Saber Elsayed
Kadir Erkan
Villatoro-Tello Esaú
Mahdi Ezoji
Shaojing Fan
Yaping Fang
Chen Fei
Chong Feng
Liangbing Feng
Alessio Ferone
Francesco Ferrise
Juan Carlos Figueroa
Michele Fiorentino

Qian Fu
Hironobu Fujiyoshi
Wai-keung Fung
Liang Gao
Xiaofang Gao
Yang Gao
Yushu Gao
Zhong-Ke Gao
Dingfei Ge
Jing Ge
Giorgio Gemignani
Shaho Ghanei
Saameh Golzadeh
Jing Gu
Smile Gu
Tower Gu
Xingsheng Gu
Jian Guan
Shi-Jie Guan
Lanshen Guo
Tiantai Guo
Weili Guo
Yinan Guo
Puneet Gupta
Haciilhan
Javad Haddadnia
Fei Han
Kyungsook Han
Meng Han
Wenting Han
Yu-Yan Han
Xin Hao
Manabu Hashimoto
Selena He
Tao He
German Hernandez
Laurent Heutte
Anush Himanshu
Huabin Hong
Lei Hou
Changjun Hu
Ke Hu
Haoqian Huang
Huali Huang
Jida Huang
Ke Huang
Lei Huang
Yea-Shung Huang
Wu-Yin Hui
Sorin Ilie
Saiful Islam
Saeed Jafarzadeh
Chuleerat Jaruskulchai
James Jayaputera
Shouling Ji
Zhiwei Ji
Hongjun Jia
Changan Jiang
He Jiang
Min Jiang
Shujuan Jiang
Ying Jiang
Yizhang Jiang
Yunsheng Jiang
Lie Jie
Xu Jie
Ning-De Jin
Wei Jin
Mingyuan Jiu
Ren Jun
Yang Kai
Hee-Jun Kang
Olesya Kazakova
Ondrej Kazik
Mohebbi Keyvan
Amar Khoukhi
Hong-hyun Kim
One-Cue Kim
Taeho Kim
Wooyoung Kim
Ogaard Kirk
Duangmalai Klongdee
Kunikazu Kobayashi
Toshiaki Kondo
Kitti Koonsanit
Takashi Kuremoto
Baeguen Kwon
Hebert Lacey
Qixun Lan
Jose A. Fernandez Leon
Bingnan Li
Bo Li
Chen Li
Dalong Li
Fuhai Li
Hui Li
Jianqing Li
Jianxing Li
Jingfei Li
Juan Li
keling Li
Qingfeng Li
Qinghua Li
Shang Li
Wei Li
Xiangyang Li
Xiaodi Li
Xiaoguang Li
Yunqi Li
Jing Liang
Xinwu Liang
Gumei Lin
Jian Lin
Yong Lin
Chenbin Liu
Chih-Chin Liu
Huai-Jen Liu
James Liu
Jin-Xing Liu
Li Liu
Liangxu Liu
Qing Liu
Xiaoming Liu
Yijian Liu
Yufeng Liu
Yuhang Liu
Zhe Liu
Alfredo Liverani
Francesco Longo
SiowYong Low
Xingjia Lu
Zhen Lu
Junfeng Luo
Durak-Ata Lutfiye
Jun Lv

Chuang Ma
Lan Ma
Wencai Ma
Xiaotu Ma
Xiaoxiao Ma
Shingo Mabu
Sakashi Maeda
Mohammad-Javad Mahmoodabadi
Guoqin Mai
Swanirbhar Majumder
Mario Manzo
Antonio Maratea
Erik Marchi
Hunny Mehrotra
Geethan Mendiz
Giovanni Merlino
Hyeon-Gyu Min
Saleh Mirheidari
Akio Miyazaki
Raffaele Montella
Tsuyoshi Morimoto
Saeed Mozaffari
Lijun Mu
Tarik Veli Mumcu
Francesca Nardone
Ken Nguyen
Zhen Ni
Changhai Nie
Aditya Nigam
Zhijun Niu
Ryuzo Okada
Kazunori Onoguchi
Dazhao Pan
Quanke Pan
Jekang Park
Anoosha Paruchuri
Giangluca Percoco
Alfredo Pereira
Elisano Pessa
Fausto Petrella
Martin Pilat
Gibran-Fuentes Pineda
Surya Prakash
Prashan Premaratne
Miguel A. Pujana
Kang Qi
Xiangbo Qi
Pengjiang Qian
Kaijin Qiu
Ying Qiu
Chenghua Qu
Junfeng Qu
Junjun Qu
Stefanos Quartini
Muhammad Rahman
Sakthivel Ramasamy
Muhammad Ramzan
Tao Ran
Muhammad Rashid
Hamidreza Rashidy Kanan
Abdul Rauf
Angelo Riccio
Lisbeth Rodríguez
Sudha Sadasivam
Angelo Antonio Salatino
Angel Sappa
Michele Scarpiniti
Donguk Seo
Chao Shao
Haojie Shen
Yehu Shen
Bo Sheng
Fanhuai Shi
Jibin Shi
Xiutao Shi
Atsushi Shimada
Nobutaka Shimada
Ye Shuang
Jakub Smid
Jakub Smidbjunior
Jakub Smidmff
Mai Son
Bin Song
Rui Song
Yang Song
Yinglei Song
Jairo Soriano
Sotanto Sotanto
Stefano Squartini
Antonino Staiano
Hung-Chi Su
Jinya Su
Rina Su
Eng.Marco Suma
Marco Suma
Guangming Sun
Jiankun Sun
Jie Sun
Jing Sun
Sheng Sun
Xiaoyan Sun
Yu Sun
Jayasudha John Suseela
Lijing Tan
Buzhou Tang
Xinhua Tang
Xiwei Tang
Tansalg
Zhu Teng
Hongjun Tian
Tian Tian
DungLe Tien
Aruna Tiwari
Kamlesh Tiwari
Mukesh Tiwari
Minglei Tong
Ximo Torres
Joaquín Torres-Sospedra
Farzad Towhidkhah
Yao-Hong Tsai
Naoyuki Tsuruta
Gurkan Tuna
Pierpaolo Valentini
Andrey Vavilin
Tomaso Vecchi
Giuseppe Vettigli
Petra Vidnerová
Aihui Wang
Bin Wang
Chun-Hsin Wang
Fang-Fang Wang
Haili Wang
Huisen Wang

Jingchuan Wang
Jinhe Wang
Jun Wang
Ling Wang
Mingyi Wang
Qixin Wang
Sheng-Yao Wang
Shulin Wang
Xiangyu Wang
Xiao Wang
Xiaoming Wang
Xiying Wang
Yan Wang
Yichen Wang
Yong Wang
Yongcui Wang
Yunfei Wang
Zhaoxi Wang
Zi Wang
Zongyue Wang
Suparta Wayan
Wei Wei
Zhihua Wei
Zhixuan Wei
Ouyang Wen
Shengjun Wen
Chao Wu
Hongrun Wu
Jingli Wu
Weili Wu
Yonghui Wu
Qing Xia
Siyu Xia
Qin Xiao
Yongfei Xiao
Keming Xie
Minzhu Xie
Zhenping Xie
Chao Xing
Wei Xiong
Dawen Xu
Jin Xu
Jing Xu
Xiaoyin Xu
Xin Xu
Ye Xu
Yuan Xu
Zhenyu Xuan
Yu Xue
Atsushi Yamashita
Mingyuan Yan
Yan Yan
Chia-Luen Yang
Chyuan-Huei Yang
Shan-Xiu Yang
Wankou Yang
Wenqiang Yang
Yang Yang
Altshuler Yaniv
Xiangjuan Yao
Shin Yatakahashi
Tao Ye
Myeong-Jae Yi
Kai Yin
Hua Yu
Liu Yu
Wu Yu
Jinghua Yuan
Lin Yuan
Quan Yuan
Assunta Zanetti
Samir Zeghlache
Yu Zeng
Zhiyong Zeng
Chunhui Zhang
Chunjiang Zhang
Duwen Zhang
Guanglan Zhang
Guohui Zhang
Hailei Zhang
Hongyun Zhang
Jianhua Zhang
Jing Zhang
Jun Zhang
Kevin Zhang
Ming Zhang
Peng Zhang
Qiangfeng Zhang
Ruofei Zhang
Shuyi Zhang
Wenxi Zhang
Xianxia Zhang
Xiaoling Zhang
Xiujun Zhang
Yanfeng Zhang
Yifeng Zhang
Yong-Wei Zhang
Zhanpeng Zhang
Changbo Zhao
Guodong Zhao
Liang Zhao
Miaomiao Zhao
Min Zhao
Xinhua Zhao
Xu Zhao
Yue Zhao
Yunlong Zhao
Bojin Zheng
Chunhou Zheng
Huanyu Zheng
Min Zheng
Xiaolong Zheng
Xinna Zheng
Liugui Zhong
Jiayin Zhou
Linhua Zhou
Songsheng Zhou
Yinzhi Zhou
Hua Zhu
Nanli Zhu
Xuefen Zhu
Yongxu Zhu
Zhongjie Zhu
Majid Ziaratban

Table of Contents

Evolutionary Learning and Genetic Algorithms

Machine Learning Theory and Methods

Biomedical Informatics Theory and Methods

Particle Swarm Optimization and Niche Technology

Unsupervised and Reinforcement Learning

Intelligent Computing in Bioinformatics

Intelligent Computing in Finance/Banking

Intelligent Computing in Petri Nets/Transportation Systems

Intelligent Computing in Signal Processing

Intelligent Computing in Pattern Recognition

Intelligent Computing in Image Processing

Intelligent Computing in Robotics

Intelligent Computing in Computer Vision

Special Session on Biometrics System and Security for Intelligent Computing

Special Session on Bio-inspired Computing and Applications

Computer Human Interaction Using Multiple Visual Cues and Intelligent Computing

Special Session on Protein and Gene Bioinformatics: Analysis, Algorithms and Applications

A Hybrid Approach for Large Scale Causality Discovery

Zhifeng Hao[1,2,*], Jinlong Huang[1], Ruichu Cai[2], and WenWen[2]

[1] Faculty of Applied Mathematics, Guangdong University of Technology, Guangzhou, China
{mazfhao,jinlonghuang13}@gmail.com
[2] Faculty of Computer Science, Guangdong University of Technology, Guangzhou, China
cairuichu@gmail.com

Abstract. Causality discovery is one of the basic problems in the scientific field. Though many researchers are committed to find the causal relation from observational data, there are still no effective methods for the high dimensional data. In this work, we propose a hybrid approach by taking the advantage of two state of the art causal discovery methods. In the proposed method, the structure learning based methods are explored to discover the causal skeleton, and then the additive noise models are conducted to distinguish the direction of causalities. The experimental results show that the proposed approach is effective and scalable for the large scale causality discovery problems.

Keywords: Nonlinear causality, Causal Markov Assumption, additive noise model.

1 Introduction

Discovering causality from the observational data is one of the basic problems in scientific field. In recent years, the researchers are committed to find the causal relationship between variables based on the observational data [1-2, 4, 5]. However, there are still no effective methods for the high dimensional data.

In statistical learning communities, Bayesian network is a useful tool for analyzing the correlation and causality relationships between variables [1-2, 4]. But the Markov equivalence class cannot be distinguished by the score function. Therefore, it just simply constructs a probabilistic model with high likelihood, instead of finding the exact causality relationship. Though, the causal skeleton is discovered, it still cannot distinguish the direction of causalities. The additive noise model [5, 6] uses the distribution of the noises to break the symmetry between variables, which can discover the causality between variables on the low dimension ($n<8$). However, the relationships between nodes on the high dimension are more

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 1–6, 2013.

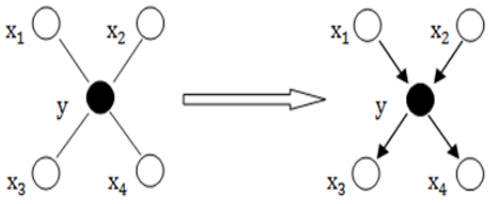
complicated. Additive noise model [5] cannot distinguish the direct causality and many causal relations may falsely be discovered. That is, both Causal Bayesian network and Additive Noise Model fail to distinguish nonlinear causality of the variables on the high dimension.

In this work, we propose a hybrid approach for large scale causality discovery, HYA in short. Basically, the structure learning based methods are explored to discover the causal skeleton, and then the additive noise models are conducted to distinguish the direction of causalities. This method will not only reduce computation complexity but also improve the accuracy of the result.

The outline of this paper is listed as follows. In section 2, provides our approach. Section 3 gives the experimental results and Section 4 concludes the paper.

2 The Proposed Method

In this section, we discuss the details of the proposed hybrid approach. For convenience, the proposed method is called HYA in the following sections.



(a) (b)

Fig. 1. HYA framework: (a) the causal skeleton of label y, (b) the causality of label y

As is shown in Figure 1 (a), in the first phase, structure learning based methods are explored to discover the causal skeleton. Unlike the GS method [20], which require an exponential number of training instances to the size of local neighborhoods of the network (PC). Though the additive noise model can distinguish nonlinear causality effectively, it cannot learn the causal skeleton. Hence, the approaches [10] are explored to discover to find the causal skeleton in the cases of less samples which keep the effectiveness when combined with the additive noise model [5]. Fig2 presents the proposed algorithm: HYA.

Input: y ,错误！未找到引用源。

Output: the causality of 错误！未找到引用源。

1. if 错误！未找到引用源。 then

2. for each $p_i \in PC(y)$错误！未找到引用源。 do

//**forward model**//

$\varepsilon_i = Residuals(p_i, y)$

$z_i = Test\ independence(p_i, \varepsilon_i)$

//**backward model**//

$\varepsilon_{fi} = Residuals(y, p_i)$

$z_{fi} = Test\ independence(y, \varepsilon_{fi})$

3. if 错误！未找到引用源。

$abs(z_i - z_{fi}) = 1$ then

Algorithm. HYA

Comparing with the ANM [5], HYA reduce the feature variable which needs to identify whether they have the potential direct causality relationship with the label variable on high dimensions in the PC phase. Therefore, it can allow HYA to discover the direct causality of y on high dimension. When using ANM [5] in the case of high dimension, the relationships between nodes on the high dimension are more complicated than the ones on a low dimension. Moreover, more nodes will be misjudged to have relation with the label variable, which not only increases the complexity of the algorithm but also leads to more redundancy. Therefore, it's hard to meet the needs of accurate recognition causal relationship by using ANM [5] on a high dimension. HYA can pick out all the feature variables which are potentially causalities of the label variable. As the cost of time is significantly decreased, it can handle causality discovery problem efficiently on a high-dimensional space.

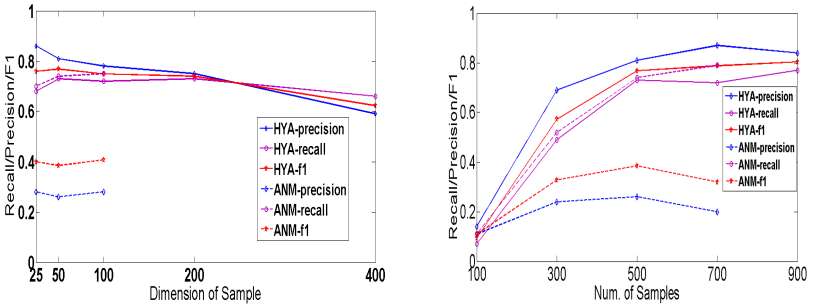
3 Experiments

In section 3.1, we introduce the setups of the experiments. In section 3.2, to illustrate the advantage of our proposal, we present and interpret the experimental results on a synthetic data set. In section 3.3, we present the experimental result on the real data sets.

3.1 Experiment Setup

In our experiments, the proposed approach is compared with other methods on both synthetic and employs the real data sets in our experiments. The KCI [9] is employed, with

conditional independence threshold of 95%. According to the rules: $\frac{Single\ parent\ of\ label\ varible}{Doubel\ parent\ of\ label\ varible}=1.5$ 错误！未找到引用源。,we constructed fixed dimensions of simulated network which is acquiescently consists of 50 dimensions by the Adjacency Matrix. For the real data sets, we use the Leukemia data set to evaluate the proposed approach. Three criteria, including precision, recall and F1 score Results on Simulated Data set.



(a) (b)

Fig. 2. (a) Different dimensions of sample. (b) Different number of sample.

As the experimental results shown above in the figure, the precision of HYA is higher than the ANM [5]. In Fig 1, we can see that when given the label node f_3, there exists a path $f_3 \to f_4 \to f_5$ 错误！未找到引用源。. Usually, what interests us more is the nodes which have direct causality of node f_3 错误！未找到引用源。, because it's more frequent for ANM to identify redundancy results like $f_3 \to f_4$ 错误！未找到 and $f_3 \to f_5$ 引用源。and $f_3 \to f_5$错误！未找到引用源。on a high dimension and also discover the wrong causality in this case. Figure 2(a) and Figure 2(b) demonstrate that HYA is less time consuming than ANM [5]. It is able to discover causality effectively for high dimensional data. However, the recall of HYA is lower than ANM [5], but the gap between them is not too large for the PC phase to depend on the generating graph connectivity. Therefore, HYA is not only able to **reduce computation complexity** but also improves the accuracy of the results, which can discover the causality of lab variable more effectively than ANM on high dimension.

3.2 Results on Real Data Set

The real dataset is used to test the effectiveness of the proposed approach. we use the Leukemia data set which is consists of 47 samples belong to ALL subtype of Leukemia and 25 samples of AML subtype under Leukemia, each of which contains the expression level of 7129 genes segments. As the label sets is discrete, we transform them into continuous data at first.

Table 1. Causality Gene Discovered on the Leukemia Data Set

Probe ID	Gene Name	Description	The report
D31886_at	KIAA0066	Homo sapiens KIAA0066 mRNA	none
X63359_at	UGT2 BIO	UGT2BIO mRNA	none
M31303_rna1 _at	Op18	Human on coprotein 18 (Op18) gene	expressed in high amounts
D63880_at	KIAA0159	Homo sapiens mRNA for KIAA0159 gene	The cause of Leukemia
M27891_at	CST3	Human Cystatin C (CST3) gene	expressed in high amounts
AFFX-DapX-5_at	AFFX-DapX-5	endogenous control	none

We try to use the HYA discover the causality between genes and Leukemia. As the results shown in Table 2, six genes are discovered. Among all the discovered genes, however, Op18 [8] has been verified to be expressed in high amounts in acute leukemia. The CST3 [9] appears as one of the most significant genes in the leukemia data which is identified by their study, so we can regard the gene as the effect of leukemia. KIAA0159 is correlated with the structural maintenance of chromosomes, it may be associated with the pathogenesis of leukemia [10]. The causality of other genes between leukemia is still not clear. Some important knowledge might be discovered if biologists pay more attention to these genes.

4 Conclusions

In this work, a hybrid approach for large-scale causality discovery, HYA, is proposed. Our results on synthetic data and real data demonstrate that our proposal is much more effective and efficient than any other existing solutions to tackle with the causality of high dimensionality. The success of HYA also reflects that structure learning based approach and the additive noise model are complementary to each other, which provide a new direction for the causality discovery problem.

References

1. Pearl, J.: Causality: Models, Reasoning, and Inference. Cambridge University Press (2000)
2. Heckerman, D., Meek, C., Cooper, G.: A Bayesian Approach to Causal Discovery. Technical Report MSR-TR-97-05, Microsoft Research (1997)
3. Hoyer, P.O., et al.: Nonlinear Causal Discovery with Additive Noise Models. In: The Twenty-Second Annual Conference on Neural Information Processing Systems (2008)
4. Shimizu, S., et al.: A Linear Non-Gaussian Acyclic Model for Causal Discovery. The Journal of Machine Learning Research 7, 2003–2030 (2006)

5. Zhang, K., et al.: Kernel-based Conditional Independence Test and Application in Causal Discovery. arXiv preprint arXiv:1202.3775 (2012)
6. Cai, R.C., Zhang, Z.J., Hao, Z.F.: BASSUM: A Bayesian Semi-supervised Method for Classification Feature Selection. Pattern Recognition 44(4), 811–820 (2011)
7. Sun, X.H., Dominik, J., Bernhard, S.: Causal Inference by Choosing Graphs with Most Plausible Markov Kernels. In: Proceeding of the 9th Int. Symp. Art. Int. and Math., Fort Lauderdale, Florida (2006)
8. Gullberg, M., Noreus, K., Brattsand, G., et al.: Purification and Characterization of A 19-kilodalton Intracellular protein. An activation-regulated Putative Protein Kinase C Substrate of T lymphocytes. J. Biol. Chein. 265, 17499–17505 (1990)
9. Tang, L.-J., Jiang, J.-H., Wu, H.-L., et al.: Variable Selection Using Probability Density Function Similarity for Support Vector Machine Classification of High-dimensional Microarray Data. Talanta 79(2), 260–267 (2009)
10. Wang, X., Gotoh, O.: Accurate Molecular Classification of Cancer Using Simple Rules. BMC Med Genomics 2(64) (2009)

Fast Wavelet Transform Based on Spiking Neural Network for Visual Images

Zhenmin Zhang, Qingxiang Wu, Zhiqiang Zhuo,
Xiaowei Wang, and Liuping Huang

College of Photonic and Electronic Engineering, Fujian Normal University, Fuzhou, China
qxwu@fjnu.edu.cn, min1011@126.com

Abstract. The functionalities of spiking neurons can be applied to deal with biological stimuli and explain complicated intelligent behaviors of the brain. Wavelet transform is a powerful time-frequency analysis tool that can efficiently compress image and extract image features. In this article, a spiking neural network combined with the ON/OFF neuron arrays associated with the human visual system is proposed to perform the fast wavelet transform for visual images. The simulation results show that the spiking neural network can preserve the key features of visual images very well.

Keywords: Spiking neural network, human visual system, fast wavelet transform, visual image.

1 Introduction

Hodgkin-Huxley Spiking Neuron Model was proposed in 1952 [1]. But if this model is applied to a large scale network, the implementation will encounter a very high computational complexity. Therefore, the simplified conductance-based integrate-and-fire model will be used for each neuron in Spiking Neuron Networks (SNNs) [2]. In the human visual system, there are various receptive fields from simple cells in the striate cortex to those of the retina and lateral geniculate nucleus [3-5]. The visual images are transferred among these neurons in the form of spiking trains through the ON or OFF pathways [6-7]. It is assumed that each neuron receives spike trains through excitatory synapse for ON neurons and through inhibitory synapse for the OFF neurons [8]. Different ON/OFF pathways are used to construct the specific network in a biological manner. On the other hand, wavelet transform can efficiently extract the key features of images [9-11]. In this paper, a SNN is proposed to mimic behaviors of spiking neurons in the human visual system for wavelet transform and extract the main features of visual images.

2 Spiking Neural Network Model for Fast Wavelet Transform

2.1 Fast Wavelet Transform

Mallat proposed fast wavelet transform (FWT) in 1987[12, 13]. The flow chart of two-dimensional FWT is shown in Fig. 1.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 7–12, 2013.

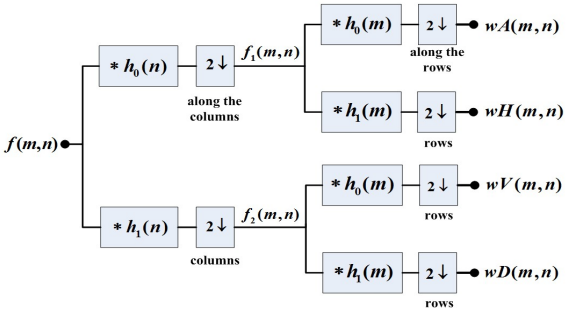


Fig. 1. Achieved 2D-FWT through the application of filter bank and down-sampling

As shown in Fig. 1, the input signal $f(m,n)$ is passed through different filters $h_0(m,n)$ and $h_1(m,n)$ and down-sampled respectively and the four signals ultimately obtained are approximate coefficient *wA*, horizontal detail *wH*, vertical detail *wV* and diagonal detail *wD* of wavelet transform.

2.2 Spiking Neural Network Model for Fast Wavelet Transform

Based on the Mallat algorithm and ON/OFF pathways mechanism in the visual system [8, 14, 15], an integrate-and-fire SNN model is proposed as shown in Fig. 2.

The dimension of the input neuron array is $M \times N$. Each pixel of the image corresponds to a receptor. Assume that $G_{m,n}(t)$ represent the gray scale of an image pixel and each photonic receptor transfers the pixel brightness to a synapse current $I_{m,n}(t)$ [16-18]. The $I_{m,n}(t)$ and the neuron potential $v_{m,n}(t)$ can be represented as follows:

$$\frac{dI_{m,n}(t)}{dt} = -\frac{1}{\tau} I_{m,n}(t) + \alpha G_{m,n}(t) \tag{1}$$

$$c\frac{dv_{m,n}(t)}{dt} = g_l(E_l - v_{m,n}(t)) + I_{m,n}(t) + I_0 \tag{2}$$

where m=1,…,M and n=1,…,N, α, τ are constants, g_l is the membrane conductance, E_l is the reverse potential, c represents the membrane capacitance and I_0 is background noise. If the membrane potential passes threshold v_{th}, then the neuron generates a spike. Let $S_{m,n}(t)$ represent the spike train generated by the neuron such as that:

$$S_{m,n}(t) = \begin{cases} 1 & if\ neuron\ (m,n)\ fires\ at\ time\ t. \\ 0 & if\ neuron\ (m,n)\ does\ not\ fire\ at\ time\ t. \end{cases} \tag{3}$$

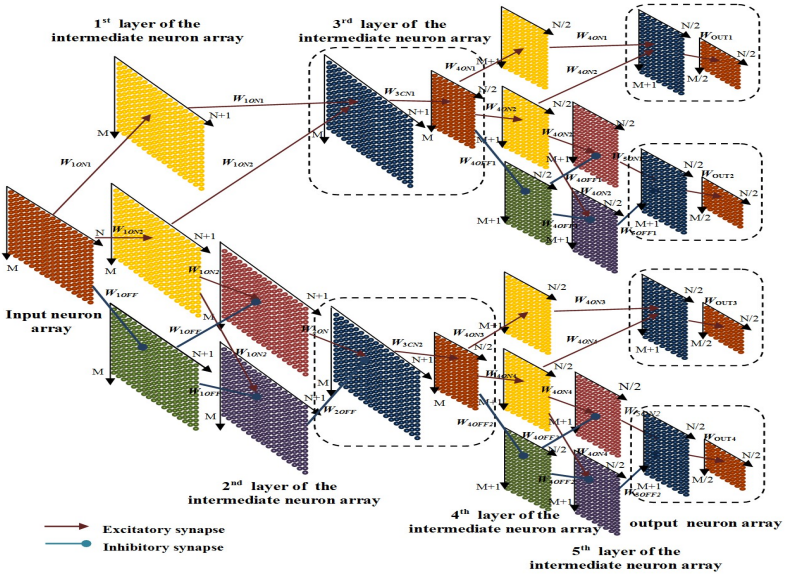


Fig. 2. Spiking neural network for fast wavelet transform

The first layer of the intermediate neuron array is composed of three $M\times(N+1)$ neuron arrays as shown in Fig. 2. First two are the ON neuron arrays 1ON1(p, q), 1ON2(p, q) and the third is the OFF neuron array 1OFF(p, q), where p=1, …, M and q=1, …, $N+1$. The convolutions of FWT are corresponding to the accumulation of different neural arrays. Assume the spike trains are transferred to the ON/OFF neuron arrays through excitatory synapses $W_{1ON1(p,q)}$ and $W_{1ON2(p,q)}$ and inhibitory synapse $W_{1OFF(p,q)}$. The synapse strength distribution can be set as follows.

$$W_{1ONi(p,k)}=a_{ON}f(p,q), \qquad W_{1OFF(p,q)}=a_{OFF}f(p,q) \tag{4}$$

where $i=\{1,2\}, 1\le p\le M$, $1\le q\le N$. if i=1, k=q, else, k=q+1. $a_{\mathrm{ON}}=1/\sqrt{2}$, $a_{\mathrm{OFF}}=-1/\sqrt{2}$. The synapse currents $I_{1ON1(p,q)}(t)$, and $I_{1OFF(p,q)}(t)$ are governed by the current constraint equation:

$$\frac{dI_{1\sigma(p,q)}(t)}{dt}=-\frac{1}{\tau}I_{1\sigma(p,q)}(t)+\sum_{p=1}^{M}\sum_{q=1}^{N+1}W_{1\sigma(p,q)}\beta_1 S_{\sigma(p,q)}(t) \tag{5}$$

where $\sigma\in\{ON,OFF\}$. $S_{\sigma(p,q)}(t)$ represent a spike train. The neuron potential in the ON/OFF array is governed by the potential constraint equation.

The intermediate second layer of the neuron array is composed of two $M\times(N+1)$ neuron arrays $2ON(p,q)$ and $2OFF(p,q)$. Each neuron receives spike trains through excitatory synapse $W_{2ON(p,q)}$ and inhibitory synapse $W_{2OFF(p,q)}$, they are set as:

$$W_{2\mathrm{ON}(p,q)} = \begin{cases} W_{1\mathrm{ON2}(p,q)} - W_{1\mathrm{OFF}(p,q)}, & if\ W_{1\mathrm{ON2}(p,q)} - W_{1\mathrm{OFF}(p,q)} > 0 \\ 0, & if\ W_{1\mathrm{ON2}(p,q)} - W_{1\mathrm{OFF}(p,q)} \le 0 \end{cases} \tag{6}$$

$$W_{2\mathrm{OFF}(p,q)} = \begin{cases} -(W_{1\mathrm{ON2}(p,q)} - W_{1\mathrm{OFF}(p,q)}), & if\ W_{1\mathrm{ON2}(p,q)} - W_{1\mathrm{OFF}(p,q)} < 0 \\ 0, & if\ W_{1\mathrm{ON2}(p,q)} - W_{1\mathrm{OFF}(p,q)} \ge 0 \end{cases} \tag{7}$$

where $1 \le p \le M$, $1 \le q \le N+1$. The synapse current and the neuron potential in the ON/OFF array are still governed by the current and potential constraint equation.

The third layer of the intermediate array is still composed of two $M\times(N+1)$ neuron arrays. Neurons in these arrays are labeled with 3CN1*(p, q) and 3CN2*(p, q). The synapses strength distribution can be calculated by the following expressions.

$$W_{3\mathrm{CN1}^*(p,q)} = W_{1\mathrm{ON1}(p,q)} + W_{1\mathrm{ON2}(p,q)}, \qquad W_{3\mathrm{CN2}^*(p,q)} = W_{2\mathrm{ON}(p,q)} - W_{2\mathrm{OFF}(p,q)} \tag{7}$$

The synapse currents are set as the following equations:

$$\frac{dI_{3CN1^*(p,q)}(t)}{dt} = -\frac{1}{\tau} I_{3CN1^*(p,q)}(t) + \sum_{p=1}^{M}\sum_{q=1}^{N+1} W_{1ON1(p,q)}\beta_2 S_{p,q}(t) + \sum_{p=1}^{M}\sum_{q=1}^{N+1} W_{1ON2(p,q)}\beta_2 S_{p,q}(t) \tag{8}$$

$$\frac{dI_{3CN2^*(p,q)}(t)}{dt} = -\frac{1}{\tau} I_{3CN2^*(p,q)}(t) + \sum_{p=1}^{M}\sum_{q=1}^{N+1} W_{2ON(p,q)}\beta_2 S_{p,q}(t) - \sum_{p=1}^{M}\sum_{q=1}^{N+1} W_{2OFF(p,q)}\beta_2 S_{p,q}(t) \tag{9}$$

where β_1, β_2 is a constant.

After the accumulation of signals, only the neurons of the even-numbered columns of the 3CN1 and 3CN2 neuron layer generate spikes, while the neurons of the odd-numbered columns do not fire. Then two new neuron arrays are obtained which are labeled with 3CN1* and 3CN2*. Synapse strength distribution can be set as follow.

$$W_{3\mathrm{CN}i(p,q)} = W_{3\mathrm{CN}i^*(p,2k)} \tag{10}$$

where i={1,2}, $k = 1, 2, \cdots, N/2$, $1 \le p \le M$, $1 \le q \le N/2$.

Thereafter, the remaining synapse strength distribution of the network can be set in a similar iteration and down-sampling manner, and eventually we will obtain four neuron array $OUT1$, $OUT2$, $OUT3$ and $OUT4$ as the bottom layer and the firing rate for these layers is calculated by the following expression:

$$r_{OUT\{j\}(m,n)}(t) = \frac{1}{T}\sum_{t}^{t+T} S_{OUT\{j\}(m,n)}(t) \tag{11}$$

where j={1,2,3,4}, $S_{OUT\{j\}(m,n)}(t)$ represent spike train generated by the output array.

3 Simulation Results

This network model is simulated by using the Euler method with a time step of 0.1 ms by Matlab. The following parameters were used in the experiments corresponding to biological neurons. v_{th} = -60 mv. E_l= -70 mv. g_l=1.0 μs/mm^2. c=8 nF/mm^2. τ = 16ms. T=400ms. α=0.02. β_1=4.3. β_2=5.1. I_0=7μA. These parameters can be adjusted to get a good quality output image.

The Lena image (512×512) is used to test the network model. Since the image exceeds the Matlab predetermined matrix dimension, therefore the image has been divided into 32×32 blocks and each block contains 16×16 pixels. Fig. 3(a)-(d) show the four coefficients of the wavelet transform obtained by Mallat method. Fig. 3(e)-(h) display the similar results obtained by SNN. In Fig. 3, the dimensions of all of the images are 8×8 and the resolution of these results is a quarter of the original image.

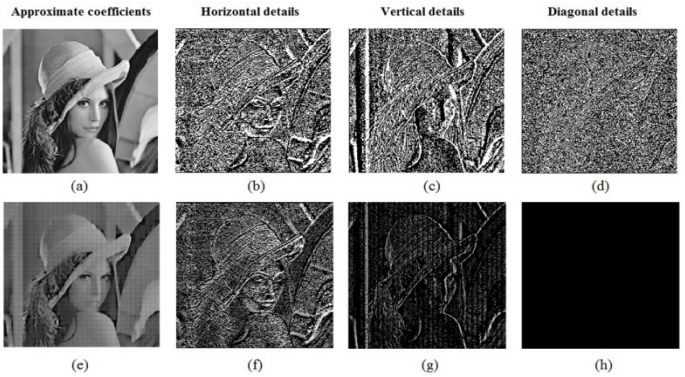


Fig. 3. Wavelet transform by Mallat method (a-d) and by SNN (e-h) of Lena

It can be seen that, although the visual image signals pass through a complex spiking neural network and lost a lot of details, but ultimately still retain all the main information and achieve the purpose of the feature extraction.

4 Discussion

In this paper, we propose an integrate-and-fire spiking neuron network combining visual ON/OFF neuron pathways and synapse current mechanism to extract features from a visual image. In the process of building the model, the accumulation between different neuron arrays are used to perform the convolutions of FWT, while the firing neurons is selected instead of the down-sampling algorithm. The simulation results show that the SNN is able to perform FWT. The key information can be obtained when the visual image signals pass through a complex spiking neural network.

Acknowledgements. The authors gratefully acknowledge the fund from the Natural Science Foundation of China (Grant No. 61179011) and the Natural Science Foundation of Fujian Province (Grant No. 2011J01340).

References

1. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of Physiology 117(4), 500–544 (1952)
2. Muller, E.: Simulation of High-Conductance States in Cortical Neural Networks. Masters thesis. University of Heidelberg. HDKIP-03-22 (2003)
3. Masland, R.H.: The fundamental plan of the retina. Nature Neuroscience 4(9), 877–886 (2001)
4. Taylor, W.R., Vaney, D.I.: New directions in retinal research. Trends in Neurosciences 26(7), 379–385 (2003)
5. Kandel, E.R., Shwartz, J.H.: Principles of Neural Science. Edward Amold (Publishers) Ltd. (1981)
6. Demb, J.B.: Cellular mechanisms for direction selectivity in the retina. Neuron. 55(2), 179–186 (2007)
7. Nelson, R., Kolb, H.: On and Off Pathways in the Vertebrate Retina and Visual System. MIT Press, Cambridge (2003)
8. Wu, Q.X., McGinnity, T.M., Maguire, L., Ghani, A., Condell, J.: Spiking Neural Network Performs Discrete Cosine Transform for Visual Images. Emerging Intelligent Computing Technology and Applications: With Aspects of Artificial Intelligence 5755, 21–29 (2009)
9. Daubechies, I.: Ten Lectures On Wavelets. Society for Industrial and Applied Mathematics 61 (1992)
10. Chui, C.K.: An Introduction to Wavelets. Academic Press, New York (1992)
11. Liu, C.L.: A Tutorial of the Wavelet Transform (2010), `http://disp.ee.ntu.edu`
12. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation 11, 674–693 (1989)
13. Mallat, S.: A Wavelet Tour of Signal Processing. Academic Press (2008)
14. Wu, Q.X., McGinnity, T.M., Maguire, L., Belatreche, A., Glackin, B.: 2D co-ordinate transformation based on a spike timing-dependent plasticity learning mechanism. Neural Networks 21(9), 1318–1327 (2008)
15. Wu, Q.X., McGinnity, M., Maguire, L., Glackin, B., Belatreche, A.: Learning Mechanisms in Networks of Spiking Neurons. In: Chen, K., Wang, L. (eds.) Trends in Neural Computation. SCI, vol. 35, pp. 171–197. Springer, Heidelberg (2006)
16. Wu, Q., McGinnity, M., Maguire, L., Belatreche, A., Glackin, B.: Edge Detection Based on Spiking Neural Network Model. In: Huang, D.-S., Heutte, L., Loog, M. (eds.) ICIC 2007. LNCS (LNAI), vol. 4682, pp. 26–34. Springer, Heidelberg (2007)
17. Wu, Q.X., McGinnity, T.M., Maguire, L.P., Glackin, B., Belatreche, A.: Learning under weight constraints in networks of temporal encoding spiking neurons. Neurocomputing 69(16-18), 1912–1922 (2006)
18. Wu, Q.X., Cai, R., McGinnity, T.M., Maguire, L., Harkin, J.: Remembering Key Features of Visual Images based on Spike Timing Dependent Plasticity of Spiking Neurons (2009)

Further Analysis on Stability for a Class of Neural Networks with Variable Delays and Impulses*

Chang-bo Yang[1,2], Xing-wei Zhou[2], and Tao Wang[2]

[1] School of Mathematical Sciences,
University of Electronic Science and Technology of China,
Chengdu, Sichuan, 611731, P.R. China
[2] Institute of Nonlinear Analysis, Kunming University,
Kunming, Yunnan, 650214, P.R. China
Cbyang348@126.com

Abstract. This paper is concerned with the global exponential stability of equilibrium point for a kind of neural networks with time-varying delays and impulsive perturbations. By using M-matrix theory, Halanay inequality and some analysis techniques, a novel condition is obtained to ascertain the global exponential stability of these networks. The derived result improves and extends some related results in the literature. Finally, an illustrative example is provided to demonstrate the effectiveness of our theoretical results.

Keywords: Neural networks, M-matrix, Variable delays, Exponential stability, Impulses.

1 Introduction

In [1], authors proposed a general class of neural networks model, which was described by the following impulsive differential equations with constant time delays:

$$\begin{cases} \dot{x}_i(t) = -c_i x_i(t) + f_i(x_1(t),\cdots,x_n(t),x_1(t-\tau_{i1}),\cdots,x_n(t-\tau_{in})) + I_i, \\ \qquad t > t_0, \quad t \neq t_k, \quad i \in N = \{1,2,\cdots,n\}, \\ \Delta x_i(t_k) = x_i(t_k^+) - x_i(t_k^-) = J_k(x_i(t_k^-)), \quad k \in Z = \{1,2,\cdots\}. \end{cases} \tag{1}$$

By using Banach fixed point theorem, spectral theory of matrix, some results were obtained to ensure the existence of equilibrium point and its exponential stability of system (1). However, it is an idealized assumption that the delays are constant. In fact, the delays are only known to be bounded but their values are unknown and time-varying in practices. Therefore, the researches of neural networks with time-varying delays are more significant than those with constant delays [1-5]. Based on model (1),

* This research is partially supported by Chinese Universities Specialized Research Fund for the Doctoral Program (20110185110020), Education-funded Projects in Yunnan Province (2012C215) and Research Fund of Kunming University (XJ11L019, YJL11006, XJ11L027).

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 13–18, 2013.

in this paper, we shall further study the following impulsive neural networks with time-varying delays:

$$\begin{cases} \dot{x}_i(t) &= -h_i(x_i(t)) + f_i(x_1(t),\cdots,x_n(t),x_1(t-\tau_{i1}(t)),\cdots,x_n(t-\tau_{in}(t))) + I_i, \\ & t > t_0,\ \ t \neq t_k,\ \ i \in N = \{1,2,\cdots,n\}, \\ \Delta x_i(t_k) &= x_i(t_k^+) - x_i(t_k^-) = J_k(x_i(t_k)),\ \ k \in Z = \{1,2,\cdots\}, \\ x_i(s) &= \varphi_i(s),\ \ s \in [t_0-\tau, t_0], \end{cases} \tag{2}$$

where $h_i(x_i(t))$ is an appropriately behaved function at time t and $\tau_{ij}(t)$ is the transition delay, which is time-varying and bounded with $0 \le \tau_{ij}(t) \le \tau$. For simplicity, other symbols possess the same meaning as that of [1]. Moreover, we assume that

(A1) $h_i : \Re \to \Re$ is continuous and there exist constants $\rho_i > 0$ such that $(x-y)(h_i(x)-h_i(y)) \ge \rho_i (x-y)^2$ for $i \in x, y \in \Re$ and $i \in N$.

(A2) $f_i : \Re^n \times \Re^n \to \Re$ is continuous and there exist constants $\alpha_{ij}, \beta_{ij} > 0$ such that $|f_i(u,v) - f_i(\tilde{u},\tilde{v})| \le \sum_{j=1}^n \alpha_{ij} |u_j - \tilde{u}_j| + \sum_{j=1}^n \beta_{ij} |v_j - \tilde{v}_j|$ for $u, v, \tilde{u}, \tilde{v} \in \Re^n$ and $i, j \in N$.

(A3) $\Xi = \delta - \alpha - \beta$ is an M-matrix, where $\alpha = (\alpha_{ij})_{n\times n}$, $\beta = (\beta_{ij})_{n\times n}$, $p \ge 1$ and

$$\delta = diag(\delta_1, \delta_2, \cdots, \delta_n),\ \ \delta_i = p\rho_i - (p-1)\sum_{j=1}^n (\alpha_{ij} + \beta_{ij}).$$

(A4) There exists a constant vector $x^* = (x_1^*, x_2^*, \cdots, x_n^*)^T \in \Re^n$ such that

$$\begin{cases} h_i(x_i^*) &= f_i(x_1^*,\cdots,x_n^*,x_1^*,\cdots,x_n^*) + I_i,\ \ i \in N, \\ J_k(x_i^*) &= 0,\ \ i \in N,\ \ k \in Z. \end{cases} \tag{3}$$

2 Preliminaries

Lemma 1. [6] Let $S \in Z^{n\times n}$, where $Z^{n\times n}$ is a set of $n \times n$ matrices with non-positive off-diagonal elements, then S is an M-matrix if and only if there exists a positive vector $\xi = (\xi_1, \xi_2, \cdots, \xi_n)^T$ such that $S\xi > 0$ or $\xi^T S > 0$.

Lemma 2. [7, 8] Assume that $a, b \ge 0$ and $p \ge 1$, then $pa^{p-1}b \le (p-1)a^p + b^p$.

Lemma 3. [9, 10] Let a and b be constants with $b > a > 0$. If the scalar function $v(t) \in C([t_0-\tau, \beta),\ \Re^+)$ satisfies

$$D^+v(t) \le -av(t) + b\sup_{t-\tau \le s \le t} v(s), \quad t \in [t_0, \beta).$$

Then, we have $v(t) \le v(t_0)\exp\{-\varepsilon(t-t_0)\}$, where $v(t_0) = \sup_{t_0-\tau \le s \le t_0} v(s), t \in [t_0, \beta)$ and ε is the unique positive root of the equation: $\varepsilon = a - b\exp\{\varepsilon\tau\}$.

3 Main Results

Theorem 1. Under assumptions (A1-A4), furthermore, if the impulsive operators $J_k(\cdot)$ satisfy the following condition:

$$J_k(x_i(t_k)) = -\gamma_{ik}(x_i(t_k) - x_i^*), \quad \gamma_{ik} \in (0,2), \quad i \in N, \quad k \in Z. \tag{4}$$

Then there exist positive constants M, μ such that, for all $i \in N$,

$$|x_i(t) - x_i^*| \le M\exp\{-\mu(t-t_0)\}, \quad t > t_0.$$

Namely, the unique equilibrium point x^* of system (2) is global exponential stability.

Proof. In view of $J_k(x_i^*) = 0, i \in N, k \in Z$ in (4), we can know that system (2) has a unique equilibrium point from assumption (A4). Suppose that $x(t) = (x_1(t), x_2(t), \cdots, x_n(t))^T$ is an arbitrary solution of system (2). Let $y_i(t) = x_i(t) - x_i^*$, then the system (2) can be transformed into the following system:

$$\begin{cases} y_i(t) &= -[h_i(y_i(t) + x_i^*) - h_i(x_i^*)] + f_i(y_1(t) + x_1^*, \cdots, y_n(t) + x_n^*, y_1(t - \\ & \quad \tau_{i1}(t)) + x_1^*, \cdots, y_n(t - \tau_{in}(t)) + x_n^*) - f_i(x_1^*, \cdots, x_n^*, x_1^*, \cdots, x_n^*), \\ \Delta y_i(t_k) &= y_i(t_k^+) - y_i(t_k^-) = -\gamma_{ik} y_i(t_k), \quad k \in Z, \quad i \in N, \\ y_i(s) &= \varphi_i(s) - x_i^*, \quad s \in [t_0 - \tau, t_0]. \end{cases} \tag{5}$$

Consider the Lyapunov function such as $V(t) = \| y(t) \|_{\xi,\infty}^p = \xi_{i_0}^{-1} | y_{i_0}(t) |^p$, where $\| y(t) \|_{\xi,\infty} = \max_{1 \le i \le n}(\sqrt[p]{\xi_i^{-1}} | y_i(t) |)$, $\xi = (\xi_1, \xi_2, \cdots, \xi_n)^T$ is determined by the following inequality (7) and $i_0 \in N$ is the index such that $\| y(t) \|_{\xi,\infty} = \sqrt[p]{\xi_{i_0}^{-1}} | y_{i_0}(t) |$.

Case 1: when $t > t_0, t \ne t_k, k \in Z$, calculating the Dini derivative of $V(t)$ along the trajectories of the continuous part of system (5), we have

$$\begin{aligned} D^+V(t) = \xi_{i_0}^{-1} p | y_{i_0}(t) |^{p-1} sign(y_{i_0}(t)) \{ & -[h_{i_0}(y_{i_0}(t) + x_{i_0}^*) - h_{i_0}(x_{i_0}^*)] \\ & + f_{i_0}(y_1(t) + x_1^*, \cdots, y_n(t) + x_n^*, y_1(t - \tau_{i_0 1}(t)) + x_1^*, \cdots, \end{aligned}$$

$$y_n(t-\tau_{i_0n}(t))+x_n^*)-f_{i_0}(x_1^*,\cdots,x_n^*,x_1^*,\cdots,x_n^*)\}.$$

From assumptions (A1-A2), Lemma 2 and simple computation, we have

$$D^+V(t)\le[-p\rho_{i_0}+(p-1)\sum_{j=1}^{n}(\alpha_{i_0j}+\beta_{i_0j})]\xi_{i_0}^{-1}\mid y_{i_0}(t)\mid^p]$$

$$+\sum_{j=1}^{n}\alpha_{i_0j}\xi_{i_0}^{-1}\xi_j(\xi_j^{-1}\mid y_j(t)\mid^p)+\sum_{j=1}^{n}\beta_{i_0j}\xi_{i_0}^{-1}\xi_j(\xi_j^{-1}\mid y_j(t-\tau_{i_0j}(t))\mid^p)$$

$$\le -aV(t)+b\sup_{t-\tau\le s\le t}V(s), \tag{6}$$

where

$$a=p\rho_{i_0}-(p-1)\sum\nolimits_{j=1}^{n}(\alpha_{i_0j}+\beta_{i_0j})-\sum\nolimits_{j=1}^{n}\alpha_{i_0j}\xi_{i_0}^{-1}\xi_j, b=\sum\nolimits_{j=1}^{n}\beta_{i_0j}\xi_{i_0}^{-1}\xi_j.$$

Since Ξ is an M-matrix, it follow from (A3) and Lemma 1 that there exists a positive constant vector $\xi=(\xi_1,\xi_2,\cdots,\xi_n)^T$ such that, for all $i\in N$,

$$p\rho_i\xi_i-(p-1)\sum_{j=1}^{n}(\alpha_{ij}+\beta_{ij})\xi_i-\sum_{j=1}^{n}\alpha_{ij}\xi_j-\sum_{j=1}^{n}\beta_{ij}\xi_j>0. \tag{7}$$

Equivalently, we have $p\rho_i-(p-1)\sum_{j=1}^{n}(\alpha_{ij}+\beta_{ij})-\sum_{j=1}^{n}\alpha_{ij}\xi_i^{-1}\xi_j>\sum_{j=1}^{n}\beta_{ij}\xi_i^{-1}\xi_j>0$ for all $i\in N$. This implies that $a>b>0$. By Lemma 3, it follows from (6) that

$$V(t)\le V(t_0)\exp\{-\varepsilon(t-t_0)\}$$

for $t>t_0, t\ne t_k, k\in Z$. Note that $V(t)\ge(\sqrt[p]{\xi_i^{-1}}\mid y_i(t)\mid)^p$ for all $i\in N$, we have

$$\mid x_i(t)-x_i^*\mid\le\sqrt[p]{\xi_{\max}}\sup_{s\in[t_0-\tau,t_0]}\parallel\varphi(s)-x^*\parallel_{\xi,\infty}\exp\{-\mu(t-t_0)\} \tag{8}$$

for $t>t_0, t\ne t_k, k\in Z, i\in N$, where $\xi_{\max}=\max\{\xi_1,\xi_2,\cdots,\xi_n\}$ and $\mu=\frac{\varepsilon}{p}$.

Case 2: when $t>t_0, t=t_k, k\in Z$, it follows from (4) that

$$\mid x_i(t_k^+)-x_i^*\mid=\mid x_i(t_k)+J_k(x_i(t_k))-x_i^*\mid=\mid x_i(t_k)-\gamma_{ik}(x_i(t_k)-x_i^*)-x_i^*\mid$$

$$=\mid 1-\gamma_{ik}\parallel x_i(t_k)-x_i^*\mid\le\mid x_i(t_k)-x_i^*\mid. \tag{9}$$

Note that $e^{-\varepsilon t}$ is a continuous function on $\Re$, it follows from ((8)-(9)) that

$$|x_i(t_k^+)-x_i^*| \le |x_i(t_k)-x_i^*| \le \sqrt[p]{\xi_{\max}}\sup_{s\in[t_0-\tau,t_0]}\|\varphi(s)-x^*\|_{\xi,\infty}\exp\{-\mu t_k\}$$
$$= \sqrt[p]{\xi_{\max}}\sup_{s\in[t_0-\tau,t_0]}\|\varphi(s)-x^*\|_{\xi,\infty}\exp\{-\mu t_k^+\}.$$

Combining *Case 1* and *Case 2*, we obtain that, for all $i\in N$,

$$|x_i(t)-x_i^*| \le M\exp\{-\mu(t-t_0)\},\quad t>t_0,$$

where $M=\sqrt[p]{\xi_{\max}}\sup_{s\in[t_0-\tau,t_0]}\|\varphi(s)-x^*\|_{\xi,\infty}$. This completes the proof. □

4 An Numerical Example

Consider the following recurrent neural networks as a special case of system (2):

$$\begin{cases} \dot{x}_i(t) &= -c_i x_i(t)+\sum_{j=1}^{2}[a_{ij}f_j(x_j(t))+b_{ij}f_j(x_j(t-\tau_{ij}(t)))]+I_i,\quad t\neq t_k,\\ \Delta x_i(t_k) &= x_i(t_k^+)-x_i(t_k^-)=J_k(x_i(t_k)),\quad k\in Z,\quad i\in\{1,2\},\\ x_i(s) &= \varphi_i(s),\quad s\in[-\tau,0]. \end{cases} \tag{10}$$

Clearly, system (10) with or without impulses has been extensively investigated in [2, 9-13]. For notational simplicity, we denote $\Xi = D-(|A|+|B|)L$, where $A=(a_{ij})_{2\times 2}$, $B=(b_{ij})_{2\times 2}$, $L=diag(l_1,l_2)$, $D=diag(d_1,d_2)$ $d_i = pc_i-(p-1)\sum_{j=1}^{2}(|a_{ij}|+|b_{ij}|)l_j$, $i=1,2$, $p\ge 1$. Moreover, take $f_j(x)=|x|$, $\tau_{ij}(t)=|1.5\sin(t)|$ for $i,j=1,2$, $p=4$ and

$$C=\begin{pmatrix}15 & 0\\ 0 & 13\end{pmatrix},\quad A=\begin{pmatrix}3 & 2\\ -2 & 4\end{pmatrix},\quad B=\begin{pmatrix}3 & -2\\ 3 & 5\end{pmatrix},\quad I=\begin{pmatrix}4.5\\ 1.5\end{pmatrix}.$$

It is easy to see that the assumptions (A1-A3) are satisfied. Then, solving equations (3) by MATLAB (7.0), we obtain that the system (10) has a unique equilibrium point $(0.5,0.5)^T$. In addition, for the impulsive part, we assume that

$$J_k(x_i(t_k))=-\gamma(x_i(t_k)-0.5),\quad \gamma\in(0,2),\quad i\in\{1,2\},\quad k\in Z.$$

From Theorem 1, we conclude that the unique equilibrium point $(0.5,0.5)^T$ of system (10) is global exponential stability. However, let $p=1,2$ in Ξ, respectively, one can check that

$$\min_{1\le i\le 2}\{c_i - \sum_{j=1}^{2} |a_{ji}|\} = 7 \not> \max_{1\le i\le 2}\{\sum_{j=1}^{2} |b_{ji}|\} = 7$$

and

$$\min_{1\le i\le 2}\{2c_i - \sum_{j=1}^{2} (|a_{ij}| + |b_{ij}| + |a_{ji}|)\} = 6 \not> \max_{1\le i\le 2}\{\sum_{j=1}^{2} |b_{ji}|\} = 7.$$

Hence, Theorem 3.1 and 3.2 in [13] are invalid for system (10). By similar analysis, the main results in [2, 9-12] are also ineffective for system (10) without impulses.

References

1. Xia, Y., Wong, P.: Global Exponential Stability of A Class of Retarded Impulsive Differential Equations With Applications. Chaos Solit. Fract. 39, 440–453 (2009)
2. Cao, J.: A Set of Stability Criteria for Delayed Cellular Neural Networks. IEEE Trans. Circuits Syst. I 48, 494–498 (2001)
3. Ozcan, N., Arik, S.: An Analysis of Global Robust Stability of Neural Networks with Discrete Time delays. Phys. Lett. A 359, 445–450 (2006)
4. Faydasicok, O., Arik, S.: Robust Stability Analysis of A Class of Neural Networks with Discrete time Delays. Neural Networks, 29–30, 52–59 (2012)
5. Huang, Y., Zhang, H., Wang, Z.: Multistability and Multiperiodicity of Delayed Bidirectional Associative Memory Neural Networks with Discontinuous Activation Functions. Appl. Math. Comput. 219, 899–910 (2012)
6. Horn, R., Johnson, C.: Topics in Matrix Analysis. Cambridge University Press, London (1991)
7. Yang, C., Huang, T.: New Results on Stability for A Class of Neural Networks with Distributed Delays and Impulses. Neurocomputing 111, 115–121 (2013)
8. Xia, Y., Huang, Z., Han, M.: Exponential *p*-stability of Delayed Cohen-Grossberg-type BAM Neural Networks with Impulses. Chaos Solit. Fract. 38, 806–818 (2008)
9. Zhou, D., Cao, J.: Globally Exponential Stability Conditions for Cellular Neural Networks with Time-varying Delays. Appl. Math. Comput. 131, 487–496 (2002)
10. Zhou, D., Zhang, L., Cao, J.: On Global Exponential Stability of Cellular Neural Networks with Lipschitz-continuous Activation Function And Variable Delays. Appl. Math. Comput. 151, 379–392 (2004)
11. Cao, J., Wang, J.: Wang, Global Asymptotic Stability of A General Class of Recurrent Neural Networks with Time-varying Delays. IEEE Trans. Circuits Syst. I 50, 34–44 (2003)
12. Huang, H., Cao, J.: On Global Asimptotic Stability of Recurrent Neural Networks with Time-varying Delays. Appl. Math. Comput. 142, 143–154 (2003)
13. Ahmad, S., Stamova, I.: Global Exponential Stability for Impulsive Cellular Neural Networks with Time-varying Delays. Nonlinear Analysis. TMA 69, 786–795 (2008)

A New Result of Periodic Oscillations for a Six-Neuron BAM Neural Network Model

Chunhua Feng[1] and Yuanhua Lin[2]

[1] College of Mathematical Science, Guangxi Normal University, Guilin, Guangxi, P.R. China, 541004
chfeng@mailbox.gxnu.edu.cn
[2] Department of Mathematics, Hechi University, Yizhou, Guangxi, P.R. China, 546300
lyh4773@163.com

Abstract. This paper discusses the existence of periodic solutions in a six neurons BAM network model. By means of Chafee's criterion of limit cycle, some sufficient conditions to guarantee the existence of periodic solutions for the system are provided. Computer simulations verify the correctness of the results.

Keywords: BAM network model, equilibrium, periodic solution.

1 Introduction

It is known that BAM neural network models as well as such models with delays have been investigated by many authors [1-9]. In [1], the authors studied the local stability and local Hopf bifurcation on a simplified BAM neural network with two delays. Yu and Cao discussed the stability and Hopf bifurcation on a four-neuron BAM neural network with delays [2]. Zhang et al. investigated the multiple Hopf bifurcations for a symmetric BAM neural network model with delay [5]. Xu et al. investigated the nontrivial periodic solutions bifurcating from local Hopf bifurcation of the six-neuron system [6]. In this paper, we discuss the following system:

$$\begin{cases} u_1'(t) = -\mu_1 u_1(t) + c_{11} f_{11}\left(u_4(t-\tau_4)\right) + c_{12} f_{12}\left(u_5\left(t-\tau_5\right)\right) + c_{13} f_{13}\left(u_6(t-\tau_6)\right), \\ u_2'(t) = -\mu_2 u_2(t) + c_{21} f_{21}\left(u_4(t-\tau_4)\right) + c_{22} f_{22}\left(u_5\left(t-\tau_5\right)\right) + c_{23} f_{23}\left(u_6(t-\tau_6)\right), \\ u_3'(t) = -\mu_3 u_3(t) + c_{31} f_{31}\left(u_4(t-\tau_4)\right) + c_{32} f_{32}\left(u_5\left(t-\tau_5\right)\right) + c_{33} f_{33}\left(u_6(t-\tau_6)\right), \\ u_4'(t) = -\mu_4 u_4(t) + c_{41} f_{41}\left(u_1(t-\tau_1)\right) + c_{42} f_{42}\left(u_2\left(t-\tau_2\right)\right) + c_{43} f_{43}\left(u_3(t-\tau_3)\right), \\ u_5'(t) = -\mu_5 u_5(t) + c_{51} f_{51}\left(u_1(t-\tau_1)\right) + c_{52} f_{52}\left(u_2\left(t-\tau_2\right)\right) + c_{53} f_{53}\left(u_3(t-\tau_3)\right), \\ u_6'(t) = -\mu_6 u_6(t) + c_{61} f_{61}\left(u_1(t-\tau_1)\right) + c_{62} f_{62}\left(u_2\left(t-\tau_2\right)\right) + c_{63} f_{63}\left(u_3(t-\tau_3)\right). \end{cases} \tag{1}$$

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 19–24, 2013.

By means of Chafee's limit cycle criterion [10]: If system (1) has a unique unstable equilibrium point, all solutions of the system are bounded, then system (1) will generate a limit cycle, namely a periodic solution. Our restrictive conditions are weaker than the conditions in [6].

2 Preliminaries

Assume that (C1) For $i=1,2,\ldots,6,\ j=1,2,3, f_{ij}(u)$ are continuous bounded monotone increasing (or decreasing) activation functions. Let $\left|f_{ij}(u)\right| \le L$ for all $u \in R.$ constant $\mu_i > 0, f_{ij}(0)=0$, and

$$\lim_{u\to 0} \frac{f_{ij}(u)}{u} = \gamma_{ij} > 0 \ (or < 0) \tag{2}$$

The linearized system of (1) is the follows:

$$\begin{cases} u_1'(t) = -\mu_1 u_1(t) + a_{11}u_4(t-\tau_4) + a_{12}u_5(t-\tau_5) + a_{13}u_6(t-\tau_6), \\ u_2'(t) = -\mu_2 u_2(t) + a_{21}u_4(t-\tau_4) + a_{22}u_5(t-\tau_5) + a_{23}u_6(t-\tau_6), \\ u_3'(t) = -\mu_3 u_3(t) + a_{31}u_4(t-\tau_4) + a_{32}u_5(t-\tau_5) + a_{33}u_6(t-\tau_6), \\ u_4'(t) = -\mu_4 u_4(t) + a_{41}u_1(t-\tau_1) + a_{42}u_2(t-\tau_2) + a_{43}u_3(t-\tau_3), \\ u_5'(t) = -\mu_5 u_5(t) + a_{51}u_1(t-\tau_1) + a_{52}u_2(t-\tau_2) + a_{53}u_3(t-\tau_3), \\ u_6'(t) = -\mu_6 u_6(t) + a_{61}u_1(t-\tau_1) + a_{62}u_2(t-\tau_2) + a_{63}u_3(t-\tau_3). \end{cases} \tag{3}$$

Where $a_{ij} = c_{ij}\gamma_{ij}\ (i=1,2,\ldots,6,\ j=1,2,3)$. The matrix form of system (3) is the following:

$$U'(t) = -\mu U(t) + AU(t-\tau) \tag{4}$$

Where $U(t)=(u_1(t),u_2(t),\cdots,u_6(t))^T$, $U(t-\tau)=(u_1(t-\tau_1),u_2(t-\tau_2),\cdots,u_6(t-\tau_6))^T$, $\mu = diag(\mu_1,\mu_2,\ldots,\mu_6)$.

Lemma 1. If the determinant of matrix $B = A+\mu$ for all given values is not equal to zero, then system (1) has a unique equilibrium point.

Proof. The linearization of system (1) around $u=0$ is (3). Noting that $f_{ij}(u)$ are monotone increasing bounded continuous activation functions. Hence, if system (3) has a unique equilibrium point which implies that system (1) also has a unique

equilibrium point. An equilibrium point $U^* = \left[u_1^*, u_2^*, \ldots, u_6^*\right]^T$ is the solution of the following algebraic equation

$$\mu U^* + AU^* = 0 \tag{5}$$

If U^* and V^* are two equilibrium points of system (5), then we have

$$\left(\mu + A\right)\left(U^* - V^*\right) = 0 \tag{6}$$

Noting that B is a nonsingular matrix, implying that $U^* - V^* = 0$, and hence $U^* = V^*$. Obviously, this equilibrium point is exactly the zero point.

Lemma 2. Each solution of system (1) is bounded.

Proof. Since $\left|f_{ij}\left(u\right)\right| \le L$ for all $u \in R$, from (1) we get

$$\frac{d\left|u_i\left(t\right)\right|}{dt} = -\mu\left|u_i\left(t\right)\right| + N_i, \left(i = 1, 2, \ldots, 6\right) \tag{7}$$

Where $N_i = \left(\left|c_{i1}\right| + \left|c_{i2}\right| + \left|c_{i3}\right|\right)L\left(i = 1, 2, \ldots, 6\right)$. Thus, when $t \ge 0$, we have

$$\left|u_i\left(t\right)\right| \le \left|u_i\left(0\right)\right|e^{-t} + N_i\left(1 - e^{-t}\right)\left(i = 1, 2, \ldots, 6\right) \tag{8}$$

This means that each solution of system (1) is bounded.

3 Periodic Oscillation Analysis

Let $\tau_* = \min\left\{\tau_1, \tau_2, \ldots, \tau_6\right\}, \mu_* = \min\left\{\mu_1, \mu_2, \ldots, \mu_6\right\}$. We know that if the unique equilibrium point is unstable, then it is still unstable when delay is increasing for a time delay system.

Theorem 1. Suppose that system (3) (or 4) has a unique equilibrium point. Let $\alpha_1, \alpha_2, \ldots, \alpha_6$ be the eigenvalues of the matrix A. If there is at least one α_j such that $-\mu_j + \mathrm{Re}\,\alpha_j > 0, j \in \left\{1, 2, \ldots, 6\right\}$, then the unique equilibrium point of system (3) is unstable, implying that system (1) generates periodic oscillations.

Proof. We first consider the case that $\tau_1 = \tau_2 = \cdots = \tau_6 = \tau_*$ in system (3), the matrix form as follows

$$U'\left(t\right) = -\mu U\left(t\right) + AU\left(t - \tau_*\right) \tag{9}$$

The characteristic equation associated with (9) given by

$$\prod_{i=1}^{6}\left(\lambda+\mu_i-\alpha_i e^{-\lambda\tau_*}\right)=0 \tag{10}$$

Noting that there exists some α_j such that $-\mu_j+\mathrm{Re}\,\alpha_j>0$, so consider the equation

$$\lambda+\mu_i-\alpha_i e^{-\lambda\tau_*}=0 \tag{11}$$

Equation (11) implies that there exists a positive real part of λ under the condition $-\mu_j+\mathrm{Re}\,\alpha_j>0$. Therefore, the trivial solution of system (9) is unstable. Because increasing time delay cannot change the instability of the equilibrium point. So for any time delays in system (1), the trivial solution is unstable. According to Chafee's criterion, system (1) has a periodic oscillation.

Theorem 2. Suppose that system (3) (or 4) has a unique equilibrium point. Assume that there exists a suitably large positive constant K such that

$$K+\mu_*-\|A\|e^{-K\tau_*}>0,\ \mu_*<\|A\| \tag{12}$$

where $\|A\|=\max_i\sum_{j=1}^{6}|a_{ij}|$. Then the unique equilibrium point of system (3) is unstable, implying that system (1) has a periodic solution.

Proof. The same as Theorem 1 we consider system (9) in the case that $\tau_1=\tau_2=\cdots=\tau_6=\tau_*$. So we have

$$\frac{dV}{dt}\le-\mu_* V(t)+\|A\|V(t-\tau_*) \tag{13}$$

Where $V(t)=\sum_{i=1}^{6}|u_i(t)|$. The characteristic equation associated with (13) given by

$$\lambda=-u_*+\|A\|e^{-\lambda t} \tag{14}$$

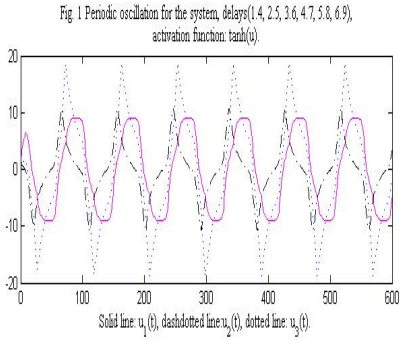
It is sufficient to show that the characteristic equation (14) has a real positive root under the condition (12). Noting that (14) is a transcendental equation. The characteristic values may be infinitely many complex numbers. We claim that there exists a real positive root for (14). Let

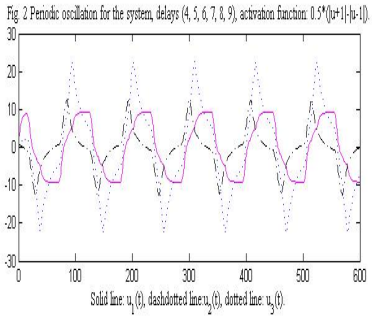
$$h(\lambda)=\lambda+\mu_*-\|A\|e^{-\lambda\tau*} \tag{15}$$

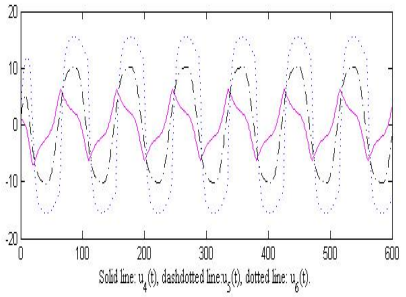
Then $h(\lambda)$ is a continuous function of λ. Since $\mu_*<\|A\|$, then $h(0)=\mu_*-\|A\|<0$, and $h(K)=K+\mu_*-\|A\|e^{-K\tau_*}>0$. Therefore, there exists $\overline{\lambda}\left(0<\overline{\lambda}<K\right)$ such that $h(\overline{\lambda})=0$ based on the continuity of $h(\lambda)$. In other words, $\overline{\lambda}$ is a real positive characteristic value of system (15). Hence, the trivial solution of (9) is unstable. Since increasing time delay in a system cannot change the instability of the equilibrium point. So for any time delays in system (3), the trivial solution is unstable. According to the criterion of the Chafee's limit cycle, system (3), thus system (1) has a periodic solution.

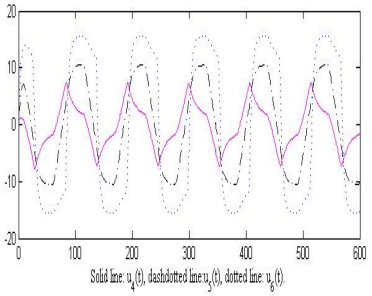
4 Computer Simulations

In system (1), we first choose the activation function $\tanh(u)$, thus $\gamma_{ij}=1(i=1,2,..,6, j=1,2,3)$. $\mu_1=0.25, \mu_2=0.06, \mu_3=0.07, \mu_4=0.08, \mu_5=0.15, \mu_6=0.25;$ $c_{11}=1.5, c_{12}=0.3, c_{13}=0.5, c_{21}=0.7, c_{22}=-1, c_{23}=0.2, c_{31}=0.8, c_{32}=0.65, c_{33}=-1.25,$









$c_{41}=-0.6, c_{42}=0.5, c_{43}=0.2, c_{51}=0.1, c_{52}=1.2, c_{53}=0.3, c_{61}=1.2, c_{62}=0.5, c_{63}=2.2.$ The eigenvalues of matrix A are: $0.6172+1.1402i$, $0.6172-1.1402i$, $-0.6172+1.1402i$, $-0.6172-1.1402i$, $-1.2558i$, $-1.2558i$. Since $-0.25+0.6172=0.3672>0$, from Theorem 1, system generates a periodic solution (see Fig.1). Then we select the activation function $\frac{1}{2}(|u+1|-|u-1|)$. The parameters are the same as figure 1 apart from the time delays. The periodic oscillation is occurred (Fig. 2).

5 Conclusion

This paper discusses a time delay BAM network model. Two theorems to guarantee the existence of periodic oscillations are derived. The restrictive conditions are weaker than the conditions of bifurcating periodic solution. Some work in the

literature has been generalized. Computer simulations are provided to demonstrate the reduced conservativeness for the time delays and parameters of the proposed results.

Acknowledgement. This research work was supported by NNSF of China (10961005), and SRF of the Education Department of Guangxi Province (201010LX463).

References

1. Cao, J., Xiao, M.: Stability and Hopf Bifurcation in A Simplified BAM Neural Network with Two Time Delays. IEEE Trans. Neural Networks 18, 416–430 (2007)
2. Yu, W., Cao, J.: Stability and Hopf Bifurcation on A Four-neuron BAM Neural Network with Delays. Phys. Lett. A 351, 64–78 (2006)
3. Sun, C., Han, M., Pang, X.: Global Hopf Bifurcation on A BAM Neural Network with Delays. Phys. Lett. A 360, 689–695 (2007)
4. Xu, C., Tang, X., Liao, M.: Stability and Bifurcation Analysis of A Six-neuron BAM Neural Network Model with Discrete Delays. Neurocomputing 74, 689–707 (2011)
5. Zhang, C., Zheng, B., Wang, L.: Multiple Hopf Bifurcations of Symmetric BAM Neural Network Model with Delay. Applied Mathematics Letters 22, 616–622 (2009)
6. Xu, C., He, X., Li, P.: Global Existence of Periodic Solutions in A Six-neuron BAM Neural Network Model with Discrete Delays. Neurocomputing 74, 3257–3267 (2011)
7. Shao, Y., Dai, B.: The Existence of Exponential Periodic Attractor of Impulsive BAM Neural Network with Periodic Coefficients And Distributed Delays. Neurocomputing 73, 3123–3131 (2010)
8. Syed Ali, M., Balasubramaniam, P.: Robust Stability of Uncertain Fuzzy Cohen-Grossberg BAM Neural Networks with Time-varying Delays. Expert Systems with Applications 36, 10583–10588 (2009)
9. Wang, Q., Cao, C., Zu, H.: A New Model Based on Grey Theory And Neural Network Algorithm for Evaluation of AIDS Clinical Trial, Adv. Comput. Math. Appl. 2, 292–297 (2013)
10. Chafee, N.: A Bifurcation Problem for A Functional Differential Equation of Finitely Retarded Type. J. Math. Anal. Appl. 35, 312–348 (1971)

A Self-Organized Fuzzy Neural Network Approach for Rule Generation of Fuzzy Logic Systems

Juan C. Figueroa-García, Cynthia Ochoa-Rey, and Jose Avellaneda-González

Universidad Distrital Francisco José de Caldas, Bogotá – Colombia
jcfigueroag@udistrital.edu.co,
{cmochoar,jaavellaneda}@correo.udistrital.edu.co

Abstract. This paper shows an algorithm for creating fuzzy logic systems from data by synchronizing its fuzzy sets and rules using a novel neuro fuzzy approach to generate rules and fuzzy sets from analyzing input data. A volatile time series example is solved and analyzed using the residuals of the model.

1 Introduction

One of the most important issues when designing an FLS is how to define its rule base. This issue increases its importance as the amount of input variables is higher, so the use of intelligent algorithms for defining the rule base of an FLS is an interesting aspect to be covered. Juang and Tsao [1] proposed a Type-2 self organized neuro-fuzzy logic system divided into two parts: a clustering algorithm for generating rules of an FLS, and a synchronization method of its fuzzy sets. In this way, our proposal is a generation method based on fuzzy neural networks using the membership degree provided by new data regarding the existent rules and backpropagation principles.

The paper is organized as follows: a first section introduces the problem. Section 2 presents come basics on fuzzy sets. Section 3 presents the proposed methodology. In section 4, an application example is solved using our proposal and finally in section 5, some concluding remarks of the proposal are shown.

2 Basics on FLSs

A fuzzy set *A* may be represented as a set of ordered pairs of a generic element *x* and its grade of membership function, $\mu_A(x)$, i.e.,

$$A = \{(x, \mu_A(x)) | x \in X\} \tag{1}$$

In this approach, *x* can be defined by multiple fuzzy sets $\{A_1, A_2, \ldots, A_m\}$, each one defined by a membership function $\{\mu_{A_1}(x), \mu_{A_2}(x), \cdots, \mu_{A_m}(x)\}$ and $\mu_A(x)$ is a measure of affinity of *x* regarding any fuzzy set *F*. Now, *A* is a *Linguistic label* which defines the sense of the fuzzy set through the word *A*. Another aspect of an FLS is the rule base, which represents the knowledge about the system. Each rule denoted by R^j relates

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 25–30, 2013.

the input variables to a consequence of its occurrence. In this way, each rule R^j can be represented as follows (See Mendel [2], and Klir and Yuan [3]):

$$R^i = IF\ x_1\ is\ A_1^i\ and \cdots and\ x_n\ is\ A_n^i, THEN\ \hat{y}\ is\ G^i;\quad i = 1, \cdots, M \tag{2}$$

where G^i is the set that represents the fuzzified output of the FLS before defuzzification. In our approach we have defined G^i as singletons, as described as follows:

$$\mu_G(x) = \begin{cases} 1 & \text{for } x \\ 0 & \text{for } X \notin x \end{cases} \tag{3}$$

The last step of an FLS is its defuzzification. When using singletons, the most used method is the *Center of sets* which is the average of all outputs, defined as follows:

$$\hat{y} = \sum_{i=1}^{M} G^i w_i \Big/ \sum_{i=1}^{M} w_i = \sum_{i=1}^{M} G^i / M \tag{4}$$

3 The Proposed Neuro-Fuzzy Approach

Our proposal is a five-layer neural network for synchronizing rules (See Figure 1) where input data is defined by $X_1, \cdots, X_n$ and the output data (a.k.a. goal data or desired response) is defined as $\hat{y}$ where X_i, $\hat{y} \in \mathbb{R}$.

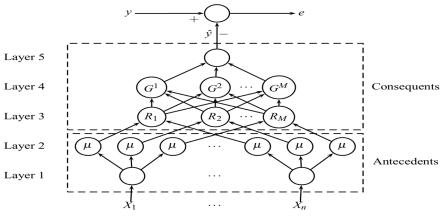


Fig. 1. Proposed neuro-fuzzy architecture

Now, the proposed structure is described as follows:

[Layer 1 - Normalization:] In this layer, all input data $x_1, \cdots, x_n$ should be normalized to any of two choices: either the interval of [-1,1] or [0,1].

[Layer 2 - Fuzzification:] This layer defines af Type-1 fuzzy set *A* for the i_{th} input, A_j^i. Usually the set A_j^i is defined with gaussian membership function.

[Layer 3 - Intersection:] Each node of this layer is a rule of the FLS, and each one uses a t-norm to compose the intersection operation. The output of a node is defined as its *activation level*, namely f^i.

[Layer 4 - Aggregation:] The number of nodes in this layer is the same as layer 3. The aggregation process consists on normalizing each f^i using the sum of all activation levels coming from layer 3, in other words:

$$G^i = f^i \Big/ \sum_i f^i \tag{5}$$

[Layer 5 - Defuzzification:] This layer simply computes the defuzzified output of the system $\hat{y}$ using the average among all aggregated values of the layer 5, as follows:

$$\hat{y} = \sum_i G^i \Big/ M \tag{6}$$

where M is the amount of rules.

3.1 Generation of Rules and Parameters of the Proposed Algorithm

The proposed neural-based algorithm for generating rules is as follows:

1. For the first input data x generate a new rule.
2. For a new input data $\breve{x}$, do:
   (a) Compute:

$$f^I(\breve{x}) = \max_{1 \leqslant i \leqslant M(t)} f^i(x) \tag{7}$$

   where $M(t)$ is the existent amount of rules at the time t.

   (b) If $f^i(\breve{x}) \leqslant \phi$, generate a new rule:

$$M(t+1) = M(t) + 1 \tag{8}$$

   where ϕ is a predefined parameter.

3. Define a new fuzzy set for each input variable $j = 1, \cdots, n$ and a new node in layer 5.

This algorithm has two key aspects: the umbral ϕ and the spread of $X_1, \cdots, X_n$.

3.2 Parameters of the Algorithm

The activation level f_i of the rule M is the degree that a rule belongs to a group (See Juang and Lin [4]). A new cluster implies a new fuzzy gaussian set[1] per input and a new node in layer 3. We have defined the consequents of the FLS as singletons with center w. Now, we propose the following equations to obtain a new group from new data:

[1] Gaussian membership functions are used due to its derivability and shape.

$$m_j^{(M+1)} = x_j \tag{9}$$

$$\delta_j^{(M+1)} = -\beta \cdot \ln(f^I) \tag{10}$$

$$w^{(M+1)} = \hat{y} \cdot K; \quad K \in [0,1] \tag{11}$$

According to Juang and Lin [4] and Juang and Tsao [1], K is a uniform random variable (See Law and Kelton [5]) which tries to explore the space of solutions.

3.3 Learning Algorithm

We use the well known backpropagation algorithm (See Wang and Mendel [6]) using gaussian membership functions for all parameters. The equations for updating the means and standard deviations of each parameter at any rule are:

$$m_j^i(t+1) = m_j^i(t) - \eta \cdot \left(\hat{y} - y \Big/ \sum_{i=1}^{M} f_i \right) \cdot (w^i(t) - \hat{y}) \cdot f_i \cdot \left(2(x_j - m_j^i(t)) \Big/ \delta_j^i(t)^2 \right) \tag{12}$$

$$\delta_j^i(t+1) = \delta_j^i(t) - \eta \cdot \left(\hat{y} - y \Big/ \sum_{i=1}^{M} f_i \right) \cdot (w^i(t) - \hat{y}) \cdot f_i \cdot \left(2(x_j - m_j^i(t))^2 \Big/ \delta_j^i(t)^3 \right) \tag{13}$$

$$w^i(t+1) = w^i(t) - \eta \cdot \left(\hat{y} - y \Big/ \sum_{i=1}^{M} f_i \right) \tag{14}$$

where η is the learning rate which can be modified as user's desires.

4 Application Example

The selected application example is the Ecopetrol shares, which presents volatility, and low correlation values in the differentiated series ($\sim<|0.2|$) which means that it may not be predicted using linear stochastic processes and/or equations. A total of 25 inputs were combined: Lagged series (SD), Sample average (MM), Bollinger bounds (BB), Momentum (Mom), Trigger signal (Trigger), Convergence/divergence of moving means (CDMM), K statistic (%K), Larry Williams statistic (%R), Relative strongness index (IFR), and Exchange rate (ROC).

Each combination has used 10 lags of SD and 50 runs of the algorithm, computing a total of 12500 experiments changing β, ϕ and η, starting from $\beta = 0.5, \phi = 0.5$ and $\eta = 0.05$. The training and validation datasets are of 247 and 37 observations.

4.1 Adapting of the Model

Two measures are used to obtain information of the model: the *RMSE* (root mean squared error), and *EMA* (absolute mean deviation) which are defined as follows:

$$RMSE = \sqrt{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \Big/ N} \tag{15}$$

$$EMA = \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{16}$$

where y_i is the desired output, $\hat{y}_i$ is the obtained output, and N is the sample size.

4.2 Obtained Results

A good model shows less RMSE and EMA values with constant mean and variance, so we have selected the best FLS which accomplishes those criteria. The best experiment is composed by SD=1, Mom=2 and ROC=3 which use 23 rules; this configuration obtained a training RMSE=0.234 and validation RMSE= 0.2614. The parameters of the algorithm are $\alpha = 0.03, \beta = 0.5$ and $\phi = 0,2$. The results of the original series for the training and validation sets are shown in Figure 3.

The best experiment spent 1000 iterations. In the original series (blue), the maximum error is $168 with a training EMA of $36.52 and a validation EMA of $42.08. Some statistical tests applied to verify the adapting of the model are shown in Table 1.

Table 1. Statistical tests

T-test on means		Walf-Wolfowitz test		Kolmogorov-Smirnov test	
Statistic	p-value	Statistic	p-value	Statistic	p-value
15.344	0.1262	0.4665	0.5271	0.5098	≈ 0

Table 1 shows a T-test contrasting that the mean of the residuals is zero. The Walf-Wolfowitz test shows that the residuals are independent and identically distributed variables, and the Kolmogorov-Smirnov test shows that the residuals of the series are normally distributed. In addition, we performed an F-test comparing data from 29/06/2010-27/12/2010 to 28/12/2010-24/06/2011 which shows a p-value of 0.1197, indicating that the variance of the residuals is constant.

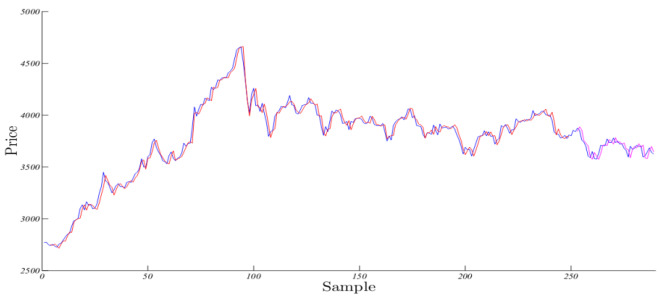


Fig. 2. Obtained results for the original series

Our proposal only generates rules for data which has not been covered by any existent rule, using ϕ as umbral. The training dataset is composed by 244 observations of each of the six input variables obtaining an appropriate model using only 23 rules, so each input data set needs about 4 rules for crossing out information to each others.

5 Concluding Remarks

The proposed approach achieves adequate results generating a reduced amount of rules. Only data which are away from existent rules generate new ones, so our proposal is sensitive to new information.

The obtained results in the application example were successful and adequate, which become our proposal as an alternative tool for identification of complex problems, such as volatile time series.

Using classical backpropagation algorithms applied to FLS and a novel algorithm for rule generation, we have obtained an adequate model with a reduced amount of rules. We encourage the reader to modify ϕ and η according to the problem.

References

1. Juang, C.F., Tsao, Y.W.: A Type-2 self-organizing neural fuzzy system and its FPGA implementation. IEEE Trans. Systems, man, and cybernetics. Part B, Cybernetics 38, 1537–1548 (2008)
2. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice Hall (2000)
3. Klir, G.J., Folger, T.A.: Fuzzy Sets, Uncertainty and Information. Prentice Hall (1992)
4. Juang, C., Lin, C.: An on-line self-constructing neural fuzzy inference network and its application. IEEE Trans. on Fuzzy Systems 6(1), 12–32 (1998)
5. Law, A., Kelton, D.: Simulation System and Analysis. Mc Graw Hill International (2000)
6. Wang, L.X., Mendel, J.M.: Back-propagation fuzzy system as nonlinear dynamic system identifiers. In: Proceedings of the FUZZ-IEEE, vol. 8, pp. 1409–1418. IEEE (1992)

Scoring Protein-Protein Interactions Using the Width of Gene Ontology Terms and the Information Content of Common Ancestors

Guangyu Cui and Kyungsook Han*

School of Computer Science and Engineering, Inha University, Incheon 402-751, Korea
khan@inha.ac.kr

Abstract. Several methods have been proposed to measure the semantic similarity of proteins. In particular, the Gene Ontology (GO) is often used to estimate the semantic similarity of proteins annotated with GO terms since it provides the largest and reliable vocabulary of gene products and their characteristics. We developed a new measure for semantic similarity of proteins involved in protein-protein interactions using the width of GO terms and the information content of their common ancestors in the GO hierarchy. A comparative evaluation of our method with other GO-based similarity measures showed that our method outperformed the others in most GO domains.

Keywords: semantic similarity, protein-protein interactions, width of GO terms.

1 Introduction

Advances in proteomics have produced a rapidly expanding and unprecedented volume of protein-protein interaction data. However, an intrinsic problem with high-throughput experimental methods is that data generated by these methods is extremely noisy. Recently various methods have been proposed to estimate the similarity of proteins involved in protein-protein interactions using the Gene Ontology (GO) terms that are annotated to proteins. Although these measures are based on GO, they use different approaches to similarity of proteins and it is not clear what measure is the best for the semantic similarity of proteins involved in protein-protein interactions.

The Gene Ontology provides the largest and reliable vocabulary of gene products and their characteristics [1, 2]. The ontology consists of a set of GO terms for each of the three domains: cellular component, molecular function, and biological process. Several similarity measures have been introduced for the relationship between GO terms. Most GO-based similarity measures use one or more of these: information content of a GO term, depth of a GO term, common ancestor of GO terms, and path length between GO terms.

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 31–36, 2013.

Unlike other similarity measures, we developed a similarity measure using the width of GO terms in addition to the information content of the lowest common ancestor of the GO terms in a directed acyclic graph of GO. We included this feature to Resnik's measure [3], and compared the new measure with the Topological Clustering Semantic Similarity (TCSS) [4], which is known as the best measure so far. We conducted a comparative evaluation of the two methods using actual protein-protein interaction (PPI) data in human and yeast. Except in the human PPI data with GO molecular function terms, our measure was better than TCSS. The rest of this paper presents the method and results of our study.

2 Related Work

Most GO-based similarity measures for proteins use one or more of the following properties of GO terms: (1) information content, (2) term depth, (3) common ancestor, and (4) path length. Similarity measures that use the depth of GO terms in the GO hierarchy assign the specificity to a GO term. Similarity measures that use the information content consider the popularity of a GO term in an annotation.

$$IC(c) = -\log[p(c)] \tag{1}$$

where p(c) is the fraction of proteins that are annotated with term c or its descendants to the proteins annotated with the root term or its descendants.

Similarity measures that use a common ancestor select common ancestors of two terms according to their properties. Information content (IC)-based measures select the proper ancestor of terms and use the information content of their common ancestor. For instance, the Maximum Informative Common Ancestor (MICA)-based approach selects the most informative common ancestor of two terms t_1 and t_2, which is the lowest common ancestor of t_1 and t_2,.

$$MICA(t_1, t_2) = \arg\max, IC(t_j) \qquad t_j \in ancestors(t_1, t_2) \tag{2}$$

Resnik's measure sim_{Res} is one of the most popular semantic similarity measures, and adopts this approach [3]. The semantic similarity between two terms t_1 and t_2 is simply the information content of the MICA.

$$sim_{Res}(t_1, t_2) = IC[MICA(t_1, t_2)] \tag{3}$$

Lin's measure sim_{Lin} [5] considers both the information content of the MICA and of the input terms.

$$sim_{Lin}(t_1, t_2) = \frac{IC[MICA(t_1, t_2)]}{IC(t_1) + IC(t_2)} \tag{4}$$

In a similar way, Jiang and Conrath's measure sim_{JC} [6] takes into account the MICA and the input terms.

$$sim_{JC}(t_1, t_2) = 1 - IC(t_1) + IC(t_2) - 2 * IC[MICA(t_1, t_2)] \tag{5}$$

Another measure called sim_{GIC} [7] is based on IC, but instead of focusing only on the lowest common ancestor of a pair of terms, it considers all the common ancestors of two sets A and B of GO terms.

$$sim_{GIC}(t_1,t_2)=\frac{\sum_{t\in(GO(A)\cap GO(B))}IC(t)}{\sum_{t\in(GO(A)\cup GO(B))}IC(t)} \tag{6}$$

where GO(X) is a set of terms within X and all their ancestors in the GO hierarchy. In general, measures that are based on common ancestors collect all the ancestors of terms, and then evaluate the overlap between them [4, 8].

Unlike previous approaches, similarity measures based on the path length [9] are correlated to the length of a path connecting two terms. For two pairs of GO terms with the same path length, the semantic distance between specific GO terms is smaller than that between less specific GO terms [10].

3 Data Acquisition and Method

Three types of data were used in this study.

1. Ontology data. Ontology data was obtained from the Gene Ontology database [11] (dated March 2010) containing 31,382 ontology terms subdivided into 2,689 cellular components, 18,545 biological processes and 8,688 molecular function terms.
2. GO annotation data. Annotations for GO terms were downloaded from the Gene Ontology database for yeast (dated February 2010) [12] and human (dated August 2010) [13]. We used the annotation data that includes the electronically inferred annotations (IEA+) because it showed a better result than the data without IEA.
3. Protein-protein interaction data. We obtained the positive and negative data of protein-protein interactions (PPI) of TCSS for comparative purpose.

The more detailed classification of concepts, the higher the semantic similarity between their decedents. Thus, we define the semantic similarity as follows.

$$sim_{Width}(t_1,t_2)=\frac{Width(MICA(t_1,t_2))}{Width(GO)} \tag{7}$$

MICA (t_1, t_2) represents the Maximum Informative Common Ancestor, Width (MICA (t_1, t_2)) represents the number of direct children of MICA (t_1, t_2), and Width (GO) is the maximum number of direct children in the GO hierarchy. Our similarity measure for terms t_1 and t_2 is a weighted sum of $Sim_{Width}(t_1, t_2)$ and Resnik's measure.

$$sim_{ours}(t_1,t_2)=\alpha * sim_{Res}(t_1,t_2)+\beta * sim_{Width}(t_1,t_2) \tag{8}$$

where $\alpha+\beta=1$. Let P and Q be the sets of GO terms annotated to gene products A and B, respectively. Then, MICA (A, B) and the semantic similarity between gene products A and B are defined by equations 9 and 10, respectively. The only difference between $Sim_{Width}(t_1, t_2)$ and $Sim_{Width}(A, B)$ is that MICA (t_1, t_2) in $Sim_{Width}(t_1, t_2)$ is changed to MICA (A, B) in $Sim_{Width}(A, B)$ since gene products A and B may be annotated with multiple GO terms.

$$MICA(A,B)=\arg\max, IC\left[MICA(t_p,t_q)\right] \qquad t_p\in P \text{ and } t_q\in Q \tag{9}$$

$$sim_{ours}(A,B) = \alpha * IC[MICA(A,B)] + \beta * \frac{Width[MICA(A,B)]}{Width(GO)} \quad (10)$$

Table 1 shows the area under the ROC curve (AUC) for the human the human PPI data with the electronically inferred annotations (IEA+). As α decreases, the result for the cellular component ontology became worse. The results shown in this paper were obtained with α=0.8 and β=0.2.

Table 1. Area under the ROC curve for the human protein-protein interaction PPI data with the electronically inferred annotations (IEA+)

	Cellular Component	Biological Process	Molecular Function
α=0.9 β=0.1	0.825	0.923	0.845
α=0.8 β=0.2	0.825	0.923	0.845
α=0.7 β=0.3	0.822	0.923	0.845
α=0.6 β=0.4	0.821	0.923	0.845

4 Results and Discussion

TCSS divides the GO graph into sub-graphs and scores PPI higher if participating proteins belong to the same sub-graph as compared to if they belong to different sub-graphs [4]. Since TCSS achieved better results than many other methods including Resnik's measure, we compared our method to TCSS with respect to AUC.

4.1 Yeast Protein-Protein Interaction Dataset

Positive and negative yeast PPI datasets were used to evaluate our method and TCSS. Our method uses only one measure to compute semantic similarity, but TCSS uses both the maximum (MAX) and best-match average (BMA). Since MAX outperforms BMA, we compared our method with MAX of TCSS. We tested with all annotations, including the electronically inferred annotations (IEA+). Figure 1 and Table 2 show that in the yeast PPI dataset our method was better than TCSS, especially for the cellular component ontology.

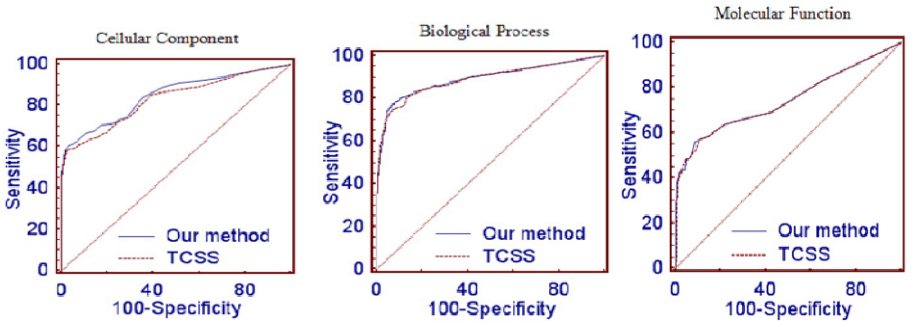


Fig. 1. ROC curves in the yeast PPI data for the cellular component, biological process and molecular function ontologies with the electronically inferred annotations (IEA+)

Table 2. Area under ROC curves for the yeast protein-protein interaction dataset with IEA+

	Cellular Component	Biological Process	Molecular Function
Our method	0.848	0.892	0.747
TCSS	0.830	0.890	0.744

4.2 Human Protein-Protein Interaction Dataset

We conducted a similar test on the human PPI dataset. Our method was better than TCSS for the cellular component and biological process ontologies, but slightly worse for the molecular function ontology (Table 3 and Figure 2).

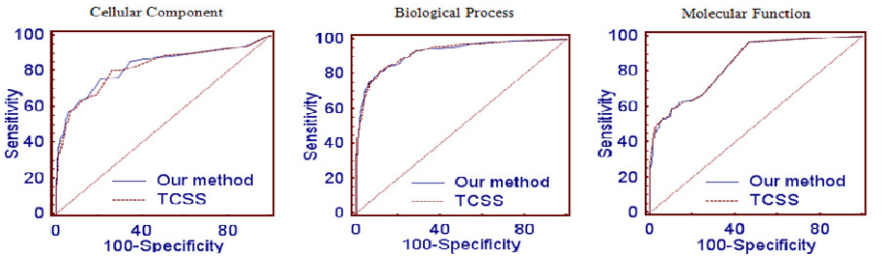


Fig. 2. ROC curves in the human PPI data for the cellular component, biological process and molecular function ontologies with the electronically inferred annotations (IEA+)

Table 3. Area under ROC curves for the human protein-protein interaction dataset with IEA+

	Cellular Component	Biological Process	Molecular Function
Our method	0.825	0.923	0.845
TCSS	0.819	0.922	0.846

5 Conclusions

Several semantic similarity measures for proteins have been proposed using the Gene Ontology. In this paper we presented a new similarity measure for proteins by incorporating the width of GO terms into Resnik's measure [3]. The width of GO terms has not been used in previous methods to assess protein similarity or protein-protein interactions. We compared our method to the Topological Clustering Semantic Similarity (TCSS) [4] using the data protein-protein interactions in yeast and human. Our method was better than TCSS for all GO ontologies except for the molecular function ontology of human. In particular, our method performed much better than TCSS for the cellular component ontology. A limitation of the method is that it cannot be applied to proteins annotated with GO terms in different ontologies. However, it is useful to assess the reliability of protein-protein interactions and predict new interactions between proteins annotated with GO terms in the same ontology.

Acknowledgements. This research was supported by the Basic Science Research program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A3011982).

References

1. Baclawski, K., Niu, T.: Ontologies for Bioinformatics (Computational Molecular Biology). The MIT Press, Cambridge (2005)
2. Harris, M.A., Clark, J., Ireland, A., et al.: The gene ontology (go) database and informatics resource. Nucleic Acids Research 32, D258–D261 (2004)
3. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: 14th Int. Joint Conf. Artificial Intelligence, pp. 448–453. Morgan Kaufmann Publishers, San Francisco (1995)
4. Jain, S., Bader, G.D.: An improved method for scoring protein-protein interactions using semantic similarity within the gene ontology. BMC Bioinformatics 11, 562 (2010)
5. Lin, D.: An Information-theoretic Definition of Similarity. In: 15th Int. Conf. Machine Learning, pp. 296–304. Morgan Kaufmann Publishers, San Francisco (1998)
6. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Int. Conf. Research in Computational Linguistics, Tapei, Taiwan, pp. 19–33 (1997)
7. Pesquita, C., Faria, D., Couto, F.M.: Measuring coherence between electronic and manual annotations in biological databases. In: ACM Symposium on Applied Computing, pp. 806–807 (2009)
8. Yu, H., Jansen, R., Stolovitzky, G., Gerstein, M.: Total ancestry measure: quantifying the similarity in tree-like classification, with genomic applications. Bioinformatics 23, 2163–2173 (2007)
9. Al-Mubaid, H., Nagar, A.: Comparison of four similarity measures based on GO annotations for Gene Clustering. Report no. 3. In: IEEE Symp. Computers and Communications, pp. 531–536 (2008)
10. Zhang, S.M., Chen, J.W., Wang, B.Y.: The research of semantic similarity algorithm consideration of multi-factor ontology-based in access control. In: Int. Conf. Computer Application and System Modeling, pp. v3-538–v3-542 (2010)
11. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. 25, 25–29 (2000)
12. Saccharomyces Genome Database (2010), `http://downloads.yeastgenome.org/`
13. The UniProt Consortium: The Universal Protein Resource (UniProt) in 2010. Nucleic Acids Research 38, D142–D148 (2010)

Database of Protein-Nucleic Acid Binding Pairs at Atomic and Residue Levels

Byungkyu Park[1], Hyungchan Kim[2], Sangmin Lee[1], and Kyungsook Han[3,*]

[1] Institute for Information and Electronics Research
[2] Department of Chemistry
[3] School of Computer Science and Engineering
Inha University, Incheon, South Korea
khan@inha.ac.kr

Abstract. As the number of structures of protein-DNA complexes and protein-RNA complexes that have been solved has been increased substantially for the past few years, a large amount of structure data is available at several resources. However, the information on the binding sites between protein and nucleic acid is not readily available from the structure data, which consist mostly of the three-dimensional coordinates of the atoms in the complexes. A few databases constructed recently provide the information on protein-nucleic acid interface, but most of them provide binding sites on either side (protein or nucleic acid) rather than binding pairs on both sides. This paper presents a new database of protein-nucleic acid binding pairs at various levels. The binding pairs identified from an extensive analysis of protein-DNA complexes and protein-RNA complexes will provide a valuable resource for studying protein-nucleic acid interactions. The database is available at http://bclab.inha.ac.kr/dbbp.

Keywords: protein-RNA interaction, protein-DNA interaction, hydrogen bond.

1 Introduction

Interaction of proteins with other molecules plays an important role in many biological activities. For example, DNA-binding proteins may activate or repress the expression of a target gene [1], so protein-DNA interactions are critical for DNA replication, transcription and gene regulations in general. Protein-RNA interactions are also involved in many biological activities in living cells. For example, tRNAs bind to aminoacyl-tRNA synthetases to properly translate the genetic code into amino acids [2], and ribonucleoprotein particles (RNPs) bind to RNA in the post-transcriptional regulation of gene expression [3]. Therefore, identification of amino acids involved in DNA/RNA binding or (ribo)nucleotides involved in amino acid binding is critical for understanding of the mechanism of gene regulations.

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 37–42, 2013.

For the past years the number of structures of protein-DNA complexes and protein-RNA complexes that have been solved has been increased, and several databases have been developed to provide information on protein-RNA or protein-DNA interactions. However, the information on the binding sites between protein and nucleic acid is not readily available from the structure data, which consist mostly of the three-dimensional coordinates of the atoms in the complexes. A recent database called the Protein-RNA Interface Database (PRIDB) [4] provides the information on protein-RNA interfaces by showing interacting amino acids and ribonucleotides in the primary sequences. However, it does not provide the information on the interacting partners of the amino acids and ribonucleotides in protein-RNA interfaces.

In this study we performed extensive analysis of the structures of protein-RNA complexes and protein-DNA interactions complexes and built a database called DBBP (**D**ata**B**ase of **B**inding **P**airs in protein-nucleic acid interactions). The database shows hydrogen-bonding interactions between proteins and nucleic acids at various levels, which is not readily available in any other databases, including the Protein Data Bank (PDB) [5]. The binding pairs of hydrogen interactions provided by the database will help researchers determine DNA (or RNA) binding sites in proteins and protein binding sites in DNA or RNA molecules. It can also be used as a valuable resource for developing a computational method aiming at predicting new binding sites in proteins or nucleic acids. The rest of the paper presents the architecture and interface of the database.

2 Materials and Methods

2.1 Protein-RNA Complexes and Protein-DNA Complexes

The protein-DNA complexes and protein-RNA complexes determined by X-ray crystallography were extracted from PDB. As of February, 2013 there were 2,568 protein-DNA complexes and 1,355 protein-RNA complexes in PDB. After selecting complexes with a resolution of 3.0 Å or better, 2,138 protein-DNA complexes (called the DS1 data set) and 651 protein-RNA complexes (the DS2 data set) remained.

2.2 Criteria for Binding Sites in Protein-Nucleic Acid Interactions

Different studies [4, 6-8] have used slightly different criteria for a binding site in protein-nucleic acid interactions. For example, in BindN [9] and RNABindR [10, 11] an amino acid with an atom within a distance of 5 Å from any other atom of a nucleotide was considered to be an RNA-binding amino acid.

As for the criteria for a binding site between proteins and nucleic acids, we use hydrogen bond (H-bond), which is stricter than the distance criteria. The positions of hydrogen atoms (H) were inferred from the surrounding atoms since hydrogen atoms are invisible in purely X-ray-derived structures. H-bonds between proteins and nucleic acids were identified by finding all proximal atom pairs between H-bond donors (D) and acceptors (A) that satisfy the following the geometric criteria: (1) the contacts with the donor-acceptor (D-A) distance < 3.9 Å, (2) the hydrogen-acceptor

(H-A) distance < 2.5 Å, (3) the donor-hydrogen-acceptor (D-H-A) angle > 90º and H-A-AA angle > 90º, where AA is an acceptor antecedent. If there is no H-bond within a protein-nucleic acid complex, we removed the complex from the data sets of DS1 and DS2. As a result, we obtained 2,068 protein-DNA complexes (DS3) and 637 protein-RNA complexes (DS4).

As an example, Figure 1 shows three H-bonds between Cytosine (C8) and Threonine (Thr224) in a protein-RNA complex (PDB ID: 4F3T) [12]. In protein-RNA interactions, N3, O2, O2', O3', O4', O5', OP1 and OP2 of Cytosine can act as a hydrogen acceptor and N3, N4, O2' and O3' of Cytosine can act as a hydrogen donor. OG1 and O of Threonine can act as a hydrogen acceptor and OG1 and N of Threonine can act as s hydrogen donor. In this example, C8 is the 8th nucleotide in RNA chain R and Thr224 is the 224th amino acid in protein chain A. O2' of C8 donates hydrogen to OG1 of Thr224, OG1 of Thr224 donates hydrogen to O2' of C8, and OG1 of Thr224 donates hydrogen to O3' of C8. Figure 2 shows the structure of the protein-RNA complex (PDB ID: 4F3T).

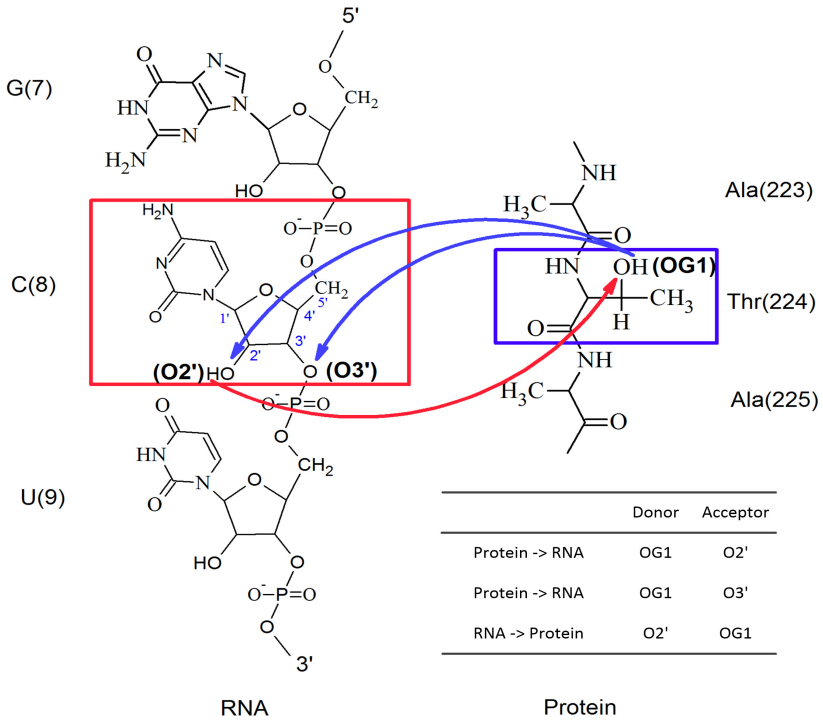


Fig. 1. Three H-bonds between Cytosine (C8) and Threonine (Thr224) of a protein-RNA complex (PDB ID: 4F3T). O2' of C8 donates hydrogen to OG1 of Thr224. OG1 of Thr224 donates hydrogen to O2' of C8 and OG1 of Thr224 donates hydrogen to O3' of C8.

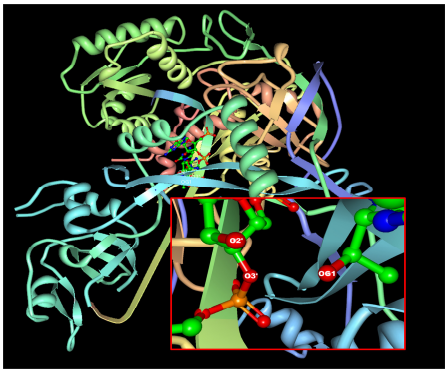


Fig. 2. The structure of protein-RNA complex (PDB ID: 4F3T). The enlarged box shows three H-bonds between Cytosine and Threonine. O2' donates hydrogen to OG1. OG1 donates hydrogen to O2' and O3'.

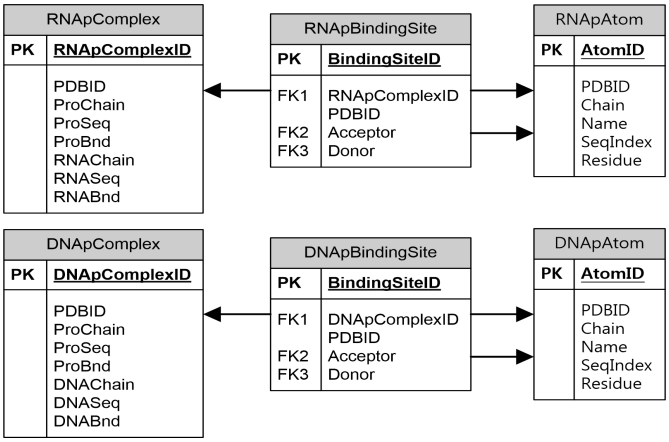


Fig. 3. The database schema

2.3 Database Schema

Figure 3 shows the database schema. The tables for protein-DNA and protein-RNA complexes (DNApComplex, RNApComplex) have columns of PDB ID, protein sequence (ProSeq), protein bond (ProBnd), DNA/RNA sequence (DNASeq, RNASeq), and DNA/RNA bond (DNABnd, RNABnd). In the DNA/RNA bond and protein bond, the '+' symbol represents a binding site and '−' represents a non-binding site.

The BindingSite tables (DNApBindingSite, RNApBindingSite) have columns of BindingSiteID (primary key), ComplexID (foreign key), PDBID, Acceptor, and Donor. The Acceptor and Donor columns of the BindingSite tables have a foreign key from two atom tables (DNApAtom, RNApAtom). The atom tables have columns, that are PDB ID, chain, atom name, sequence index (SeqIndex), and residue.

3 Results and Discussion

3.1 Hydrogen Bonds in Protein-Nucleic Acid Interactions

We identified H-bonds from 2,068 protein-DNA complexes (DS3) and 637 protein-RNA complexes (DS4) using HBPLUS [13] with the H-bond criteria: $\overline{DA} < 3.9$ Å, $\overline{HA} < 2.5$ Å, $\angle DHA > 90°$. There are a total of 77,947 H-bonds in protein-RNA complexes and 44,955 H-bonds in protein-DNA complexes. In the 77,947 H-bonds of protein-RNA complexes, there are 18,151 hydrogen acceptors and 59,796 hydrogen donors in amino acids. In the 44,955 H-bonds of protein-DNA complexes, there are 3,657 hydrogen acceptors and 41,298 hydrogen donors in amino acids. In the 77,947 H-bonds of protein-RNA complexes, there are 59,796 hydrogen acceptors and 18,151 hydrogen donors in RNAs. In the 44,955 H-bonds of protein-DNA complexes, there are 41,298 hydrogen acceptors and 3,657 hydrogen donors in DNAs.

If an atom of RNA acts as a hydrogen acceptor, an atom of protein should be a hydrogen donor. Hence, the number of RNA acceptors (59,796) is the same as the number of protein donors (59,796) and the number of RNA donors (18,151) is the same as the number of protein acceptors (18,151). Likewise, the number of DNA acceptors (41,298) is the same as the number of protein donors (41,298), and the number of DNA donors (3,657) is the same as the number of protein acceptors (3,657).

3.2 Web Interface of the Database

DBBP shows binding pairs at various levels, from the atomic level to the residue level. When it shows detailed information on H-Bonds, it shows the donors and acceptors of each H-bond. A same type of atom can play a role of hydrogen donor or acceptor depending on the context. We generated XML files for binding sites of protein-DNA/RNA complexes. Users of the database can access the XML file via PDB ID.

4 Conclusion

From an extensive analysis of the structure data of protein-DNA complexes and protein-RNA complexes extracted from PDB, we have identified hydrogen bonds (H-bonds). Analysis of the huge amount of structure data for H-bonds is labor-intensive, yet provides useful information for studying protein-nucleic acid interactions. The protein-DNA complexes contain 44,955 H-bonds, which have 3,657 hydrogen acceptors (HA) and 41,298 hydrogen donors (HD) in amino acids, and 41,298 HA and 3,657 HD in nucleotides. The protein-RNA complexes contain 77,947 H-bonds, which have 18,151 HA and 59,796 HD in amino acids, and 59,796 HA and 18,151 HD in nucleotides. Using the data of H-bonding interactions, we developed a database called DBBP (**D**ata**B**ase of **B**inding **P**airs in protein-nucleic acid interactions). DBBP provides the detailed information of H-bonding interactions between proteins and

nucleic acids at various levels. Such information is not readily available in any other databases, including PDB, but will help researchers determine DNA (or RNA) binding sites in proteins and protein binding sites in DNA or RNA molecules. It can also be used as a valuable resource for developing a computational method aiming at predicting new binding sites in proteins or nucleic acids. The database is available at `http://bclab.inha.ac.kr/dbbp`.

Acknowledgments. This research was supported by the Basic Science Research program (2012R1A1A3011982) and in part by the Key Research Institute program (2012-0005858) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology.

References

1. Simicevic, J., Deplancke, B.: DNA-centered approaches to characterize regulatory protein-DNA interaction complexes. Molecular Biosystems 6, 462–468 (2010)
2. Moras, D.: Aminoacyl-tRNA synthetases. Current Opinion in Structural Biology 2, 138–142 (1992)
3. Varani, G., Nagai, K.: RNA recognition by RNP proteins during RNA processing. Annual Review of Biophysics and Biomolecular Structure 27, 407–445 (1998)
4. Lewis, B.A., Walia, R.R., Terribilini, M., Ferguson, J., Zheng, C., Honavar, V., Dobbs, D.: PRIDB: a protein-RNA interface database. Nucleic Acids Research 39, D277–D282 (2011)
5. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The Protein Data Bank. Nucleic Acids Research 28, 235–242 (2000)
6. Allers, J., Shamoo, Y.: Structure-based analysis of Protein-RNA interactions using the program ENTANGLE. Journal of Molecular Biology 311, 75–86 (2001)
7. Norambuena, T., Melo, F.: The Protein-DNA Interface database. BMC Bioinformatics 11 (2010)
8. Spirin, S., Titov, M., Karyagina, A., Alexeevski, A.: NPIDB: A database of Nucleic Acids-Protein Interactions. Bioinformatics 23, 3247–3248 (2007)
9. Wang, L.J., Brown, S.J.: BindN: a web-based tool for efficient prediction of DNA and RNA binding sites in amino acid sequences. Nucleic Acids Research 34, W243–W248 (2006)
10. Terribilini, M., Lee, J.H., Yan, C.H., Jernigan, R.L., Honavar, V., Dobbs, D.: Prediction of RNA binding sites in proteins from amino acid sequence. RNA 12, 1450–1462 (2006)
11. Terribilini, M., Sander, J.D., Lee, J.H., Zaback, P., Jernigan, R.L., Honavar, V., Dobbs, D.: RNABindR: a server for analyzing and predicting RNA-binding sites in proteins. Nucleic Acids Research 35, W578–W584 (2007)
12. Elkayam, E., Kuhn, C.D., Tocilj, A., Haase, A.D., Greene, E.M., Hannon, G.J., Joshua-Tor, L.: The Structure of Human Argonaute-2 in Complex with miR-20a. Cell 150, 100–110 (2012)
13. McDonald, I.K., Thornton, J.M.: Satisfying Hydrogen-Bonding Potential in Proteins. Journal of Molecular Biology 238, 777–793 (1994)

Assessment of Protein-Graph Remodeling via Conformational Graph Entropy

Sheng-Lung Peng* and Yu-Wei Tsay

Department of Computer Science and Information Engineering
National Dong Hwa University, Hualien 974, Taiwan
slpeng@mail.ndhu.edu.tw

Abstract. In this paper, we propose a measurement for protein graph remodeling based on graph entropy. We extend the concept of graph entropy to determine whether a graph is suitable for representing a protein. The experimental results suggest that when this extended graph entropy is applied, it helps a conformational on protein graph modeling. Besides, it also contributes to the protein structural comparison indirectly if a protein graph is solid.

Keywords: Protein structural similarity, Protein graph, Graph entropy.

1 Introduction

Graph theory is now widely used in information theory, combinatorial optimization, structural biology, chemical molecule, and many other fields. Graph similarity measuring is a practical approach in various fields. When graphs are used for representing of structured objects, then the problem of measuring object similarity turns into the problem of computing the similarity of graphs [1]. Protein remodeling is another field where multiple domains within structures are considerably complicated.

For decades, many studies have been devoted on defining topological relations and notation on protein structures. A schematic description is essentially expected to describe its topology. Mathematical formulation of structural patterns helps to facilitate the composition in a polypeptide chain. A schematic description has the advantage of simplicity, which makes it possible to implement in an alternative way as graph-theoretic approach [2]. By selectively neglecting protein structural features, it has a potential to detect further homologous relationships based on various geometric methods and motivations.

2 Preliminaries

The structure of a protein can be regarded as a conformation with various local elements (helix, sheet) and forces (Van der Waal's forces, hydrogen bonds), folding into its specific characteristic and functional structure. With the help of *graph transformation*, folded polypeptide chains are represented as a graph by several

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 43–48, 2013.

mapping rules. Proteins contain complex relationships in its polymer; reaction of residues, interaction of covalent, bonding of peptides, packing of hydrophobic, are essential part in structure determination. The intention is to transform a protein structure into a graph.

2.1 Protein Remodeling

As mentioned to the protein remodeling, a study reviewed in detail of protein graph (abbreviated as P-graph) description can be found in [3]. In Table 1, we outline some categories of protein graph approach to a set of graphs, representing each specific graph rewriting and graph measuring skills. And also, it will be useful to begin with the summarized common research into the following matters:

Geometric Relation: It has been shown that the conformation of a protein structure is determined geometrically by using various constraints [4]. The most common method for protein modeling is to reserve its topological relation on graphs. From the aspect of graph theory, a simplified representation of protein structure aims attention at connectivity patterns. It helps to go into details on interacted relation within a polypeptide folding.

Chemical Relation: Comparing with geometric relationship, in chemical properties, it describes a more complicated description on protein graph model; owing to various chemical properties of amino acids, it includes electrostatic charge, hydrophobicity, bonding type, size and specific functional groups [5]. By giving values to edges and vertices in graph, each labeled component corresponds to a type of chemical relation.

Table 1. Recent studies for constructing protein graphs

Studies	Vertex set	Edge set
[6]	C_α atoms	labeled edges
[7]	DSSP[1]	attributed edges
[8]	side chains	defined by interacted energy
[9]	residues	defined by geometrical constraints
[10]	SSE[2]	labeled edges

[1]Dictionary of protein secondary structures. [2]Secondary structure elements.

2.2 Entropy

Entropy defines quantitatively an equilibria property within a system and it implies the principle of disorder from the second law of thermodynamics [11]. It is particularly important in describing how energy is applied and transferred in an isolated system. The higher the disorder, the greater the entropy of the system [12]. Similarly, this concept is also included in life. As we known, life is composed by many cells, tissues, and organs from one of the vital element -- protein. Since proteins are biochemical compounds, consisting of one or more polypeptide chains, the arrangement of protein polymers is assumed to be in a compact state, according to its backbone dihedral angles and side chain rotamers. This is so called *conformational entropy*. In general, a protein graph model should also obey the second law of thermodynamics.

For an n-object system G, asume that each object i is associated a probability P_i. Then the entropy of the system G is defined as follows [13].

$$I(G) = -\sum_{i=1}^{n} P_i \log P_i \tag{1}$$

In graph theory, the entropy of a graph is usually defined by its degree sequence. For example, we consider the cycle with four vertices, *i.e.*, C_4. The degree sequence is (2, 2, 2, 2). Thus, the P_i for each vertex v_i is 2/8=0.25. By definition, $I(C_4) = -4 \cdot 0.25 \cdot \log(0.25) = 2$.

3 Our Method

In this section, we extend the concept of graph entropy for masuring protein graphs. To demonstrate the calculation of graph entropy exemplarily, peptide chains of MHC (the Major Histocompatibility Complex) are selected as the materials for examining the utilities of graph entropy.

Usually, the entropy of a graph is defined by its degree sequence. In this case, every regular graph with the same order has the same entropy. For example, both of C_4 and K_4, the complete graph with four vertices, have the same entropy, namely, 2. However, C_4 and K_4 are different. Thus, this definition is not enough to distinguish these two graphs. It motivates this research.

For a given graph $G = (V, E)$ and two vertices u and v belonging to V, let $d(u,v)$ denote the length of the shortest path between u and v. Let $S_k(u) = \{v \mid d(u,v) = k\}$. In graph theory, $S_k(u)$ is called the *k-distance neighborhood* of u and is also called the *k-sphere* of u [14]. Let the function $f(u) = \sum(|S_i(u)|/(n-i+1))$ and $f(V)=\sum f(u)$. Assume that $V=\{v_1, v_2, ..., v_n\}$. We define Q_i for each v_i as follows.

$$Q_i = \frac{f(v_i)}{f(V)-S_1(v_i)+1} \tag{2}$$

Note that in Formula 1, $\sum P_i=1$. However, in Formula 2, $\sum Q_i$ is not necessary equal to 1. Thus, we call $I(G)$ the *extended entropy* of graph G by replacing P_i with Q_i in Formula 1. By this extension, we obtain that $I(C_4)=2.122$ but $I(K_4)=1.245$.

4 Results

In this experiment, we validate the remodeling function of P-graph by using extended graph entropy to verify stability of a given P-graph. For the P-graph construction, please refer [7]. Thus, we only concern the connectivity impact to protein structural similarity. Various types of MHC are chosen as the material for the verification of proposed method, namely, **1HDM**, **1K5N**, **2CRY**, **1VCA**, **2Q3Z**, and **1ZXQ**. MHC, as an immune system in most vertebrates, encodes for a small complex cell surface protein. It is also known for HLA (Human Leukocyte Antigen). Due to a great diversity of microbes in the environment, MHC genes widely vary their peptides through several mechanisms [15].

4.1 P-Graph Entropy Comparison

Let $G = (V, E)$ be the P-graph after remodeling from the construction proposed by [7], vertices of V in G are created according to the dictionary of protein secondary structures (DSSP). Under this metric, a protein secondary structure is represented by a single letter code, *e.g.*, H-helix (containing **G**, **H**, and **I**), T-hydrogen turn (containing **T**, **E**, and **B**), and C-coiled (containing only **C**). For controlling one variable in this experiment, let the edge set E in G be changed from a specific range.

A preliminary comparison of MHC proteins are shown in Table 2. In the table, **PID** is the protein identification number in PDB [16]. Since MHC proteins are composed by multiple polypeptide chains, there are multimeric **Domain**. Besides, the **Dens** means the density in the graph; it defines as $2|E|/(|V|(|V|-1))$ ranging from 0 to 1. **AVG** indicates the average distance within DSSP vertices; if the distance of v_i and v_j is no greater than **AVG**, then there is an edge between them. In the table, if **AVG** = 10, then +20% increases the criteria length from 10 to 12, which increases the number of edges in E. When the number of edges in G is raised, certainly, the density is also increased.

Table 2. The selected proteins with corresponding extended entropies

PID	Domain	-40%	-20%	AVG	+20%	+40%
1HDM	B	NA	3.252	3.319	3.531	3.660
Dens		0.381	0.476	0.524	0.667	0.762
1K5N	A	4.029	4.456	4.777	5.001	5.243
Dens		0.345	0.491	0.582	0.636	0.782
2CRY	A	NA	NA	4.029	NA	NA
Dens		0.667	0.667	0.667	1.000	1.000
1VCA	A	NA	3.253	3.412	3.412	3.249
Dens		0.333	0.476	0.571	0.571	0.857
2Q3Z	A	5.418	5.904	6.610	7.565	9.715
Dens		0.221	0.308	0.413	0.551	0.750
1ZXQ	A	NA	3.480	3.630	3.866	4.176
Dens		0.321	0.429	0.500	0.607	0.786

It is interesting to observe the relation between $|E|$ and $I(G)$ in the following. First, when the density in G raises, it appears that the graph G goes from sparse to dense. However, its extended entropy does not totally decrease with its density. It seems a little anomalous in this appearance. Second, the edge set in protein remodeling issue can be possibly determined from its entropy. By the definition, the P-graph G should be a connected graph, *i.e.*, once the G becomes a spanning tree, the conformation can be decided from its entropy. For instance of **1VCA**, its P-graph is not a connected graph when density is 0.333 (-40%) but its entropy is 3.253 when the density rises to 0.476 (-20%). There is considerable validity to this concept though it should be verified by further proof and experiment. Third, it seems that E is considerably related to V in graph entropy. Consider the P-graph **2CRY** as another example. If a protein remodeling function adapts a specific value on the basis of its geometrical edge, then it might be an error to assume a fixed value as a criteria. This is an essential fact to stress. It may be worth pointing out that the construction of P-graph is limited by V. Taking protein **1CXR** as an example, in PDB file, **1CXR** contains only one helix structure. Therefore, it would be unsuitable to transfer it into a one-vertex graph.

4.2 P-Graph Entropy Verification

For the purpose of validity according to the previous assumptions, a method for protein structural comparison is adapted to measure its similarity. *Graph spectra* gives an alternative solution for graph matching. It is a set of relational parameters, consisting of a characteristic polynomial and eigenvectors of its adjacency matrix or Laplacian matrix. Graph spectra quantitatively provide graph information, *e.g.*, structure, topology, connectivity. Please refer [17] for the detail. In Table 3, we list the results of protein structure remodeling matters. The field **Old** shows a remodeling based on specific value of edge length, and **New** indicates the edges in G are determined by extended entropy. Values in each column display a local and global comparison of their graph spectra. As the structural alignment method, the smaller of the value, the more similar of their similarity. If our entropy suggests a better result in protein comparison, then we simply mark **Better** denoted as "+"; otherwise, it is marked as "=" (unchanged) or "-" (worse). In summary, the extended entropy determines a better conformational graph from protein structure remodeling.

Table 3. A comparison of protein structure remodelings by proposed method

PID		1K5N	2CRY	1VCA	2Q3Z	1ZXQ
1HDM	Old	2.45|7.93	0.00|23.4	2.00|15.7	4.24|24.0	2.45|13.7
	New	1.73|7.75	0.00|21.1	2.23|13.9	2.83|23.7	2.24|12.1
	Better	+	+	+	+	+
1K5N	Old		1.00|26.6	2.23|19.6	3.32|26.9	2.65|18.0
	New		0.00|23.7	2.45|17.4	2.23|21.1	2.53|16.0
	Better		+	+	+	+
2CRY	Old			1.00|14.9	0.00|12.3	0.00|17.1
	New			1.00|12.9	0.00|34.1	0.00|14.9
	Better			+	=	+
1VCA	Old				1.41|17.7	1.00|5.39
	New				1.00|29.7	1.00|4.47
	Better				=	+
2Q3Z	Old					2.24|19.4
	New					1.41|28.6
	Better					=

4.3 Program and Environment

The environment is running under 2 Ghz PC with 512 MB of main memory with Linux-2.6.11-1.1369. The implementation is temporarily written using Bash-3.00.16(1) and Octave-3.0.0.

5 Conclusion

In this paper, we propose a benchmark to determine graph stability for protein structure remodeling based on graph entropy. With the help of this extended entropy validation, it concludes a conformational confirmation on protein structural

comparison. This graph-based approach offers a practical concept to support protein structural alignment. In the future, a labeled protein remodeling is also expected to be verified by this extended entropy formula.

Acknowledgement. This work is supported in part by the National Science Council, Taiwan, under the Grant NSC 101-2221-E-259-004

References

1. Bunke, H.: Graph Matching: Theoretical Foundations, Algorithms, and Applications. In: Proc. Vision Interface 2000, pp. 82–88 (2000)
2. Gilbert, D., Westhead, D.R., Nagano, N., Thornton, J.M.: Motif-based Searching in TOPS Protein Topology Databases. Bioinformatics 15, 317–326 (1999)
3. Vishveshwara, S., Brinda, K.V., Kannan, N.: Protein Structure: Insights from Graph Theory. Journal of The Comp. Chem. 1, 187–211 (2002)
4. Lund, O., Hansen, J., Brunak, S., Bohr, J.: Relationship between Protein Structure and Geometrical Constraints. Protein Science: A Publication of the Protein Society 5, 2217–2225 (1996)
5. Nelson, D.L., Cox, M.M.: Lehninger Principles of Biochemistry, 4th edn. Freeman (2004)
6. Huan, J., Bandyopadhyay, D., Wang, W., Snoeyink, J., Prins, J., Tropsha, A.: Comparing Graph Representations of Protein Structure for Mining Family-specific Residue-based Packing Motifs. J. Comput. Biol. 12, 657–671 (2005)
7. Hsu, C.-H., Peng, S.-L., Tsay, Y.-W.: An Improved Algorithm for Protein Structural Comparison based on Graph Theoretical Approach. Chiang Mai Journal of Science 38, 71–81 (2011)
8. Canutescu, A.A., Shelenkov, A.A., Dunbrack, R.L.: A Graph-theory Algorithm for Rapid Protein Side-chain Prediction. Protein Sci. 12, 2001–2014 (2003)
9. Samudrala, R., Moult, J.: A Graph-theoretic Algorithm for Comparative Modeling of Protein Structure. J. Mol. Biol. 279, 279–287 (1998)
10. Borgwardt, K.M., Ong, C.S., Schonauer, S., Vishwanathan, S.V.N., Smola, A.J., Kriegel, H.-P.: Protein Function Prediction via Graph Kernels. Bioinformatics 21, i47–i56 (2005)
11. Shannon, C.E.: Prediction and Entropy of Printed English. Bell Systems Technical Journal 30, 50–64 (1951)
12. Chang, R.: Physical Chemistry for the Biosciences. University Science (2005)
13. Simonyi, G.: Graph Entropy: a Survey. Combinatorial Optimization 20, 399–441 (1995)
14. Dehmer, M., Emmert-Streib, F.: Structural Information Content of Networks: Graph Entropy based on Local Vertex Functionals. Computational Biology and Chemistry 32, 131–138 (2008)
15. Pamer, E., Cresswell, P.: Mechanisms of MHC Class I – Restricted Antigen Processing. Annual Review of Immunology 16, 323–358 (1998)
16. Berman, H.M., Westbrook, J., Feng, Z., et al.: The Protein Data Bank, Nucl. Nucl. Acids Res. 28, 235–242 (2000)
17. Peng, S.-L., Tsay, Y.-W.: On the Usage of Graph Spectra in Protein Structural Similarity. Journal of Computers 23, 95–102 (2012)

A Novel Feature Selection Technique for SAGE Data Classification

K.R. Seeja

Department of Computer Science,
Jamia Hamdard University, New Delhi, India
seeja@jamiahamdard.ac.in

Abstract. Computational diagnosis of cancer from gene expression data is a binary classification problem. Serial Analysis of Gene Expression (SAGE) is a sequencing technique used for measuring the expression levels of genes. Each SAGE library contains expression levels of thousands of genes (or features). It is impossible to consider all these features for classification and also the general feature selection algorithms are not efficient with this data. In this paper, a data mining technique called *closed frequent itemset mining* is proposed for feature selection. Subsequently these selected genes or features are used for the training and testing of two well known classifiers- Extreme Learning Machine (ELM) and Support Vector Machine (SVM). The performance evaluation of ELM and SVM classifiers shows that the proposed feature selection method works well with these classifiers.

Keywords: Closed frequent itemset mining, Feature Selection, Serial Analysis of Gene Expression, Extreme Learning Machine, Support Vector Machine, Classification.

1 Introduction

Classification techniques are always assisted by feature selection techniques. Prior to classification, feature selection techniques are used to select the best features that distinguish the different classes. The general method of feature selection is to rank the features based on certain metrics denoting their classification capability and then selecting different sets of the top features starting from a smaller set. The objective of feature selection is to select the minimum number of features that can best classify the data. Closed frequent itemset mining is a data mining technique used for extracting frequent itemsets from large databases. In gene expression databases genes are the items and frequent itemsets are co-regulated genes. Closed frequent itemsets is a compressed representation of frequent itemsets. In this paper, closed frequent itemset mining is used for finding set of co-regulated genes from SAGE data and then these co-regulated genes are used as best features for classification.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 49–54, 2013.

2 Related Work

Several efforts of applying data mining techniques to SAGE data can be found in literature. Various data mining techniques such as classification [9,10,11,12], clustering [1,2] and association rule mining[3,4] were applied to SAGE data.

Classification is a data mining technique used to predict group membership for data instances. Here the data is classified into one of the predefined classes. From machine learning perspective classification is a supervised learning technique. Popular classification techniques include Bayesian Classifiers[5], Support Vector Machines[6], K-Nearest Neighbour[7], decision trees[8] and Neural Networks[8]. Classification techniques are mainly using in SAGE data analysis for classifying tumours or cancers. Xin et al[9] proposed a Naive Bayesian classifier for SAGE data classification. They also proposed event models [10] for SAGE expression profiles and then these models were used for binary as well as multiclass classification. Decision trees and support vector machines [11] were proposed to classify the SAGE data according to cell states(cancerous or normal) and tissue types. Okun and Priisalu proposed ensembles of nearest neighbours[12] for cancer classification using SAGE data .

There were many research works on proposing different feature selection techniques for selecting a subset of genes from SAGE data which can simplify the classification. Tzanis et al[13] proposed a new approach that uses frequent pattern mining to discover any associations among the expressions of genes that can assist the construction of more accurate classifiers. Genetic algorithm [14] and information gain [9] based feature selection methods were also proposed for simplifying SAGE data classification.

3 Proposed Approach

In this paper, a novel closed frequent itemset mining based feature selection method for classification is proposed . The various steps are as follows:

1. Training and testing dataset preparation
2. Feature selection using Closed Frequent Item Set Mining
3. Data classification using any classification methods like ELM or SVM on selected features

3.1 Data Set Preparation

SAGE data related to breast tissue has been identified by using the Library Finder tool available at Cancer Genome Project website (cgap.nci.nih.gov/SAGE/SAGELibrary Finder). Table 1 gives the details of these libraries.

Table 1. Breast Tissue Specific SAGE Libraries

Tissue Type	Minimum number of Records	Maximum number of Records	Number of Libraries
Normal	13614	26342	7
Cancerous	3983	31311	22

Since the library sizes are not same, they should be normalized before comparison. For normalizing the data, divide the frequency value by the total number of tags in the library and then multiplied by 300,000, the estimated number of RNAs per cell[3]. Half of the normal and cancerous libraries have been selected alternately as the training dataset. Thus the training dataset contains 12 cancerous and 4 normal libraries. The remaining 12 cancerous and 3 normal libraries of the dataset are selected as test dataset.

3.2 Feature Selection

In this paper, a different feature selection method based on Closed Frequent Itemset Mining[20] is proposed. A special data mining algorithm called GeneExpMiner[4] is used for finding the frequent itemsets. GeneExpMiner is a closed frequent itemset mining algorithm designed especially for mining SAGE data. Instead of ranking the genes individually here set of genes are ranked based on their co-expression patterns since it has been identified that a group of genes are acting together in the scenario of cancer. The training dataset prepared in section 3.1 is used for finding the best features. The dataset has to be converted into a Boolean matrix for applying GeneExpMiner. By applying the procedure described in [4] the training dataset has been converted to a Boolean matrix. This is the input to the GeneExpMiner. Another input to the GeneExpMiner is the support value. Different sets of closed frequent itemsets can be constructed with different support values. The set of co-expressed genes (closed frequent itemsets) are ranked based on the support values. The set with highest support is considered as best feature group. With 100% support, there was only one closed frequent itemset with 26 genes. The expressions of these 26 genes or features were used for training and testing the ELM and SVM classifiers.

3.3 Classification Using ELM and SVM

The final training dataset has been created by selecting the records correspond to selected features. Thus only 26 features of the training set created in section 3.1 have been selected for training the ELM classifier. Similarly final test data set has been prepared by selecting the expression values correspond to these 26 genes of the test data set created in section 3.1. If the expression value corresponds to any of the selected 26 genes is found to be missing in test data then the corresponding frequency is considered as '0'. The inputs to the ELM are the training set, test set, activation function and the number of hidden nodes. The number of hidden nodes can be found by a trial and error method starting from 1. The dataset for SVM is created by combining the training dataset and test datatest of 26 genes.

4 Implementation

GeneExpMiner is implemented in C++ and MATLAB 7.4 is used for performance analysis of ELM and SVM classifiers. Using trial and error method , optimal number of hidden layer neurons are found as '8'. The SVM classifier selects randomly the training and test data using 'hold out' cross validation. The performance of the classifiers is evaluated in terms of testing accuracy and training time. Table 2 shows the performance comparison of ELM and SVM on different kernel functions.

Table 2. Performance Comparison ELM Vs SVM

Classification Model	Kernel Function	Testing Accuracy	Training Time(seconds)
ELM	Sigmoid	1	0.0156
ELM	Radial Basis	0.5333	0.312
ELM	Sine	0.4000	0.0468
ELM	Hardlim	0.8000	0.0468
ELM	Triangular Basis	0.2000	0.0468
SVM	Linear	0.9333	0.7488
SVM	Quadratic	0.9333	0.8112
SVM	RBF	0.8000	0.6708
SVM	Polynomial	0.9333	0.7332
SVM	Multilayer Perceptron	0.9333	0.7644

The best performance of ELM was found with sigmoid kernel and that of SVM was with polynomial kernel. The best performance comparison of ELM and SVM are shown in figure 1 and figure 2.

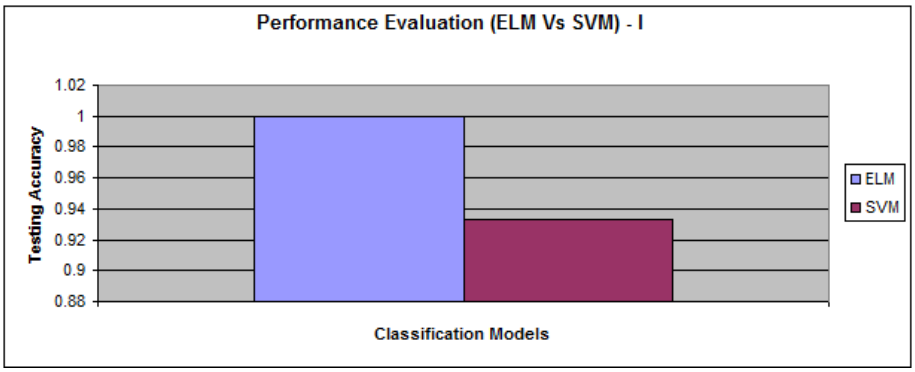


Fig. 1. Performance Comparison on Testing Accuracy

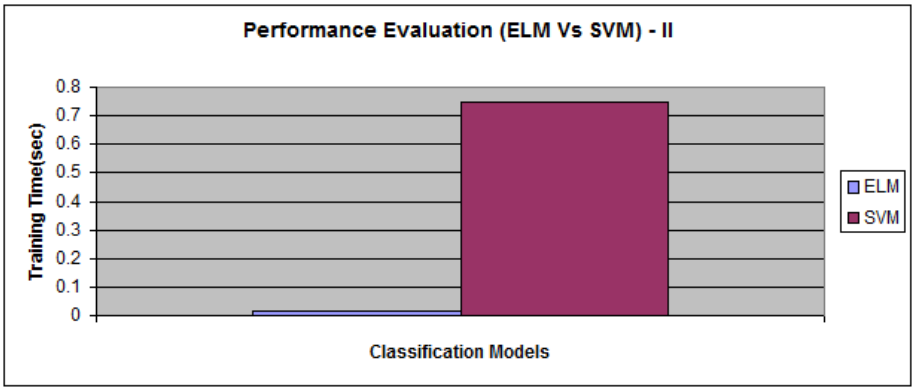


Fig. 2. Performance Comparison on Training Time

5 Conclusion

This paper proposes a novel feature selection technique based on closed frequent itemset mining. The proposed feature selection method identifies a set of co-regulated genes as top features. The SAGE data related to breast tissue is used for performance evaluation. The dataset contains the expression levels of 141611 different genes. The proposed feature selection method identifies a group of 26 co-regulated genes as top features and from the performance of ELM and SVM classifiers, it is found that knowing the expression levels of these 26 genes are enough to detect whether a breast tissue is normal or cancerous. It is also found that ELM outperforms SVM in terms of accuracy and training time.

References

1. Ng, R.T., Sander, J., Sleumer, M.C.: Hierarchical Cluster Analysis of SAGE Data for Cancer Profiling. In: Workshop on Data Mining in Bioinformatics, pp. 65–72 (2001)
2. Tzanis, G., Vlahavas, I.: Mining High Quality Clusters of SAGE Data. In: 2nd VLDB Workshop on Data Mining in Bioinformatics, Vienna, Austria (2007)
3. Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J.F., Gandrillon, O.: Strong-association-Rule Mining for Large-scale Gene-expression Data Analysis: A Case Study on Human SAGE Data. Genome Biology 3(12) (2002)
4. Seeja, K.R., Alam, M.A., Jain, S.K.: An Association Rule Mining Approach for Co-Regulated Signature Genes Identification in Cancer. Journal of Circuits, Systems, and Computers 18(8), 1409–1423 (2009)
5. Becker, B., Kohavi, R., Sommerfield, D.: Isualizing The Simple Baysian Classifier. In: Information Visualization in Data Mining and Knowledge Discovery, pp. 237–249. Morgan Kaufmann Publishers (2001)
6. Cortes, C., Vapnik, V.: Support Vector Networks. Machine Learning 20(3), 273–297 (1995)

7. Cunningham, P., Delany, S.J.: K-Nearest Neighbour Classifiers, Technical Report UCD-CSI-2007-4,March 27 (2007)
8. Han, J., Kamber, M.: Data Mining Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2006)
9. Jin, X., Xu, A., Zhao, G., Ma, J., Bie, R.: Multinomial Event Naive Bayesian Modeling for SAGE Data Classification. Springer Journal of Computational Statistics 22(11), 133–143 (2007)
10. Jin, X., Xu, A., Zhao, G., Ma, J., Bie, R.: Cancer Classification from Serial Analysis of Gene Expression with Event Models. Springer Journal of Applied Intelligence 29(1), 35–46 (2008)
11. Gamberoni, G., Storari, S.: Supervised and Unsupervised Learning Techniques for Profiling SAGE Results. In: ECML/PKDD Discovery Challenge Workshop, Pisa, Italy, pp. 121–126 (2004)
12. Okun, O., Priisalu, H.: Ensembles of Nearest Neighbour Classifiers and Serial Analysis of Gene Expression. In: SCAI 2006, Helsinki, Finland, pp. 106–113 (2006)
13. Tzanis, G., Vlahavas, I.: Accurate Classification of SAGE Data Based on Frequent Patterns of Gene Expression. ICTAI (1), 96–100 (2007)
14. Yang, C.-H., Shih, T.-M., Chuang, L.-Y.: Reducing SAGE Data Using Genetic Algorithms. International Journal of Information and Mathematical Sciences 5(4), 268–272 (2009)
15. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme Learning Machine: Theory and Applications. Neurocomputing 70, 489–501 (2006)
16. Huang, G.-B., Wang, D.H., Lan, Y.: Extreme Learning Machines: A Survey. International Journal of Machine Leaning and Cybernetics 2(2), 107–122 (2011)
17. Huang, G.-B., Zhou, H., Ding, X., Zhang, R.: Extreme Learning Machine for Regresion and Multiclass Classification. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics 42(2), 513–529 (2012)
18. Velculescu, V.E., Zhang, L., Vogelstein, B., Kinzler, K.W.: Serial Analysis of Gene Expression. Science 270, 484–487 (1995)
19. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. In: SIGMOD Conference, pp. 207–216 (1993)

Chinese Sentiment Classification Based on the Sentiment Drop Point

Zhifeng Hao*, Jie Cheng, Ruichu Cai, Wen Wen, and Lijuan Wang

[1] Faculty of Computer Science, Guangdong University of Technology Guangzhou, China
{mazfhao,googcheng,cairuichu}@gmail.com,
{wwen,ljwang}@gdut.edu.cn

Abstract. The exploding Web opinion data has the essential need for automatic tools to analyze people's sentiments in many fields. Predicting the polarity of a product review is an important work in applications such as market investigation and trend analysis. In this paper, we focus on analyzing the Chinese sentiment word strengths and the sentiment drop point. We propose a novel algorithm based on the sentiment drop point algorithm to conduct sentiment polarity assignment. It predicts the sentiment polarity by a determinative policy which involves two classifiers simultaneously. The experiments show that our approach is efficient and suited for reviews analysis in different domains.

Keywords: Sentiment drop point, Sentiment strength, Normalized Google distance.

1 Introduction

Sentiment analysis refers to analyzing subjective information in source materials with the application of text analyzing procedure, natural language processing algorithm and computational linguistic techniques. Generally, sentiment analysis consists of three stages. Firstly, the input text is split into several segments, such as sentences, and each segment is evaluated whether it is subjective or objective [1]. And then, a sentiment polarity is assigned to the subjective sentence. Finally, the object or aspect corresponding to a sentiment or an opinion may be extracted.

There has been some important work on sentiment analysis. These works can be roughly divided into three categories: based on unsupervised learning, based on supervised learning and based on lexicon. There has been some previous sentiment analysis work about unsupervised learning. Turney presents an unsupervised learning algorithm for classifying a review according to PMI (pointwise mutual information) in conjunction with two human-selected seed words ("excellent" and "poor") [2]. Zagibalov and Carroll develop a new method of automatic seed word selection for unsupervised sentiment classification of product reviews in Chinese, which avoids problems of domain-dependency [3].

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 55–60, 2013.

The supervised learning based methods focus on the sentiment strength. Pang and Lee apply a meta-algorithm to classify movie reviews to star rating [4]. Wilson et al. [5] employ supervised learning techniques to train classifiers to predict the intensity of clauses and sentences.

Compared with first two kinds of work, lexicon-based approaches have an advantage in extracting sentiment strengths. Wan proposes a lexicon-based approach to leverage reliable English resources and improve Chinese sentiment analysis [6]. Thelwall proposes the SentiStrength algorithm whose core is the sentiment word strength lexicon. The sentiment strength lexicon is coded by human judgments and it's optimized by a training procedure at the prophase [7]. In 2011, Thelwall proves that the SentiStrength2 is robust enough in the field of social web contexts [8]. Lu et al. measure the polarity strength of each adjective or phrase based on a link analysis method to generate a sentiment lexicon [9].

An improved algorithm based on the SO-CAL [10] is proposed here. We solve several drawbacks in prior work. Firstly, we can extract sentiment phrases in a unified form by our implemented program sentiPY[1]. Secondly, the Stanford Parser software[2] is used for extracting sentiment phrases and detecting negations. Thirdly we propose a propagation-based algorithm to optimize sentiment word strengths and a compound method based on the sentiment drop point algorithm to predict sentiment polarities.

The paper is organized as follows: Section 2 introduces the details of our proposed approach which is called sentiDP because the method is based on the sentiment drop point algorithm. Results and analysis on several experiments are shown in section 3. Finally, we conclude this work and point out some directions for future research in section 4.

2 The SentiDP

The Sentiment Drop Point can be used to enhance sentiment polarity assignment. For example, there is a review about iPhone 5: the device is sleek and well-built but high price, all in all it's great. We can find the sentiment drop point "great" via the summary phrase "all in all". The formal definition of Sentiment Drop Point is given as follows:

Definition 1: the Sentiment Drop Point is a sentiment part of a review or document, with which the author expresses the main sentiment.

Our workflow of sentiment classification consists of four steps: text preprocessing, sentiment word strength calculation, sentiment phrase extraction and sentiment polarity assignment. The details of four steps are described below.

Text preprocessing is an important procedure for sentiment classification. It mainly includes removing interferential clauses (like irrealistic blocking) and nature language processing.

[1] https://github.com/goog/sentiPY
[2] http://nlp.stanford.edu/software/lex-parser.shtml

2.1 Sentiment Word Strength Calculation

We create two lexicons manually: the sentiment word strength lexicon and the adverb factor lexicon. The sentiment word strength lexicon includes 2268 sentiment words scored with different grades which are scaled from -5 to +5 upon its meaning on most contexts. And the adverb factor lexicon contains eighty adverbs tagged within the region [-1, 1.5], acting as a multiplier of a sentiment word strength.

When regarding the sentiment words of a review as nodes in a graph, we can optimize the prior sentiment word strengths based on contexts by introducing normalized Google distance (NGD) [11]. NGD can measure the semantic similarity or correlation between two terms. We find adjacent sentiment word pairs in extracted phrases and apply the neighbor relationship between a pair to construct a graph then perform the propagation-based algorithm on the sentiment words graph [12]. We calculate the weight of each edge in the relationship graph by NGD.

2.2 Sentiment Phrase Extraction

We consider a sentiment phrase as a central sentiment word combined with some intensified modifiers. In this paper, all phrases are defined in the unified form:

$$phrase : \mathrm{mod}\,ifier * sentiment$$

In the above form, the expression "modifier*" is inspired by the regular expression, which stands for zero or more than one modifier.

In order to compute the sentiment phrase strength, we can implement a factorial for several intensified modifiers. We compute the strengths of phrases by the formula (1):

$$S(OP) = \prod S(\mathrm{mod}) \cdot S(sentiment) \tag{1}$$

It is needed to mention that negative words can change or even reverse the strength of a sentiment word or phrase. We'll apply the Stanford typed dependency **neg** (negation modifier) to detect negations and then perform a polarity shift.

2.3 Sentiment Polarity Assignment

Finding the sentiment drop point and assigning the polarity to the sentiment drop point is one of our explorations. We apply the following steps to find the sentiment drop point in a review:

1. A sentiment drop point can follow some summative words like "totally";
2. People are used to representing the sentiment drop point at the beginning or the ending of a review. So we can get the sentiment drop point by comparing the beginning sentiment strength with the ending one in a review;
3. If the sentiment drop point isn't found in two tries above, at last we can assume that the most strongest sentiment phrase stands for a good chance of a drop point in a review.

When inputting the phrases extracted by the sentiPY, the common method the sum of sentiment phrase strengths is more accurate on negative reviews while the sentiment drop point algorithm performs better on positive reviews. Both of them are hybrid to get a better performance for all cases. Therefore, the sentiDP a compound method, consisting of the above two algorithms.

3 Experiments and Analysis

The review data sets[3] used in the experiment are provided by Doctor Songbo Tan on and collected from online comments on the Ctrip.com website. We have employed two Aliyun[4] Elastic Compute instances to claw data and it takes about one month to get more than two millions index count items from Sogou[5].

3.1 Evaluation on the Hotel Data

In this experiment, two methods are compared with the proposed method on the hotel data set. The baseline method is to classify product reviews by counting the number of sentiment words. And in Lu's algorithm, Lu creates a sentiment strength lexicon automatically and then predicts the polarities.

As shown in Figure 1, it is obvious that our method achieves the best accuracy 80.45% on the hotel reviews data set and has a 4.125% improvement compared with the baseline method. From the results, we can conclude that our method is efficient because we have introduced sentiment word strengths and a better evaluation algorithm.

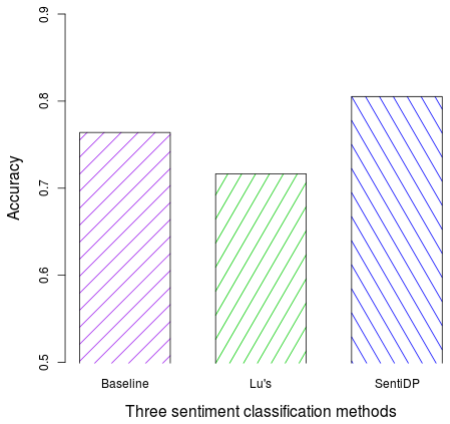


Fig. 1. Different methods on the hotel data set

[3] http://www.searchforum.org.cn/tansongbo/corpus-senti.htm
[4] http://www.aliyun.com/product/ecs/
[5] http://www.sogou.com/

3.2 Evaluation on Different Domains

In this experiment, the proposed method is compared with the two methods on two data sets. We employ a negative amplifie in the "common" method. The "common" algorithm is to aggregate all sentiment strengths in a review. The "droppoint" one depends on the sentiment drop point algorithm and predicts the polarity by the strength of the sentiment drop point in a review.

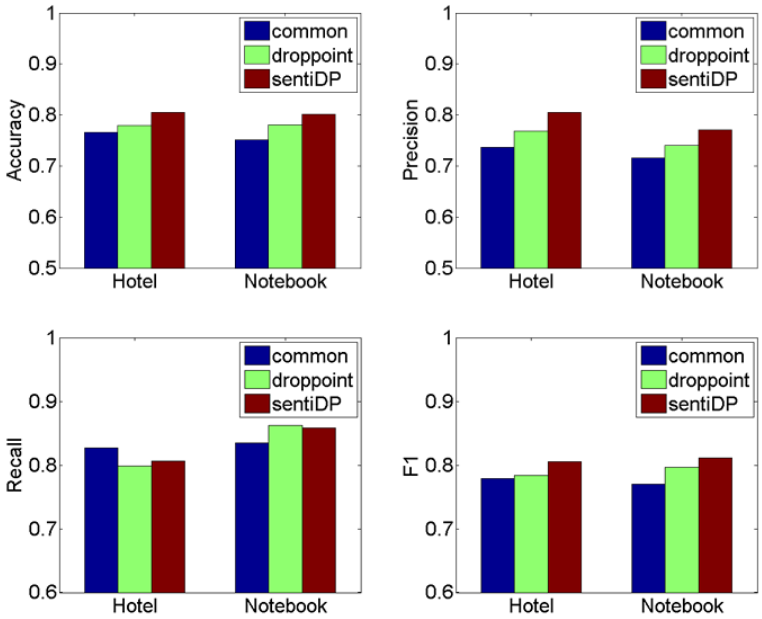


Fig. 2. Evaluations on hotel and notebook reviews

As shown in Figure 2, the results show that the SentiDP get the best accuracy, precision and F-measure on both the hotel and notebook reviews. It achieves the accuracy 80.525% and the precision 80.45% on the hotel data. In addition, it reaches the F-measure 81.12% on the notebook data. Based on that, we can make a conclusion that the sentiDP is efficient for different domain reviews.

4 Conclusions

The lexicon-based method is consistent with human intuition and interpretable. The experimental results show that the sentiDP achieves 4.125% improvement comparing to the baseline method and gets the best accuracy, precision and F-measure on the hotel and notebook reviews. The success of SentiDP also shows its applicability for different domains. The method can be further improved by getting more accurate data

from web, expanding the sentiment word strength lexicon and introducing a knowledgebase like ConceptNet[6].

References

1. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, p. 271. Association for Computational Linguistics (2004)
2. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 417–424. Association for Computational Linguistics (2002)
3. Zagibalov, T., Carroll, J.: Automatic seed word selection for unsupervised sentiment classification of Chinese text. In: Proceedings of the 22nd International Conference on Computational Linguistics, vol. 1, pp. 1073–1080. Association for Computational Linguistics (2008)
4. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. Annual Meeting-Association for Computational Linguistics 43(1), 115 (2005)
5. Wilson, T., Wiebe, J., Hwa, R.: Recognizing strong and weak opinion clauses. Computational Intelligence 22(2), 73–99 (2006)
6. Wan, X.: Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 553–561. Association for Computational Linguistics (2008)
7. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. Journal of the American Society for Information Science and Technology 61(12), 2544–2558 (2010)
8. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment strength detection for the social web. Journal of the American Society for Information Science and Technology (2012)
9. Lu, Y., Kong, X., Quan, X., Liu, W., Xu, Y.: Exploring the sentiment strength of user reviews. Web-Age Information Management, 471–482 (2010)
10. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Computational Linguistics 37(2), 267–307 (2011)
11. Cilibrasi, R.L., Vitanyi, P.M.B.: The google similarity distance. IEEE Transactions on Knowledge and Data Engineering 19(3), 370–383 (2007)
12. Zhang, J., Tang, J., Li, J.: Expert finding in a social network. Advances in Databases: Concepts. Systems and Applications, 1066–1069 (2007)

[6] http://conceptnet5.media.mit.edu/

Multi-objectivization and Surrogate Modelling for Neural Network Hyper-parameters Tuning

Martin Pilát[1] and Roman Neruda[2]

[1] Faculty of Mathematics and Physics, Charles University in Prague,
Malostranské náměstí 25, Prague, Czech Republic
Martin.Pilat@mff.cuni.cz
[2] Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod Vodárenskou věží 2, Praha
roman@cs.cas.cz

Abstract. We present a multi-objectivization approach to the parameter tuning of RBF networks and multilayer perceptrons. The approach works by adding two new objectives – maximization of kappa statistic and minimization of root mean square error – to the originally single-objective problem of minimizing the classification error of the model. We show the performance of the multi-objectivization approach on five data sets and compare it to a surrogate based single-objective algorithm for the same problem. Moreover, we compare the multi-objectivization approach to two surrogate based approaches – a single-objective one and a multi-objective one.

Keywords: Multi-objective optimization, parameter tuning, neural networks, surrogate modelling, multi-objectivization.

1 Introduction

Using evolutionary algorithms to optimize the parameters of classifiers brings several difficult challenges. First of all, most of the classifiers provide similar (or even the same) results for lots of different settings of their parameters, this leads to a search space, where most of the points (i.e. different parameter settings) yield the same error rate [13]. This problem is even more pronounced if the given classification problem has only a small number of instances. In this case, there are only a few different error rate values possible. Thus, the fitness function is piecewise constant, sometimes with only a small area of a local optima – exactly the type of fitness function which is difficult to optimize using the evolutionary algorithms.

Machine learning algorithms also require quite a long time to train (especially on larger data sets), and evolutionary algorithms often need a large number of fitness function evaluations to find a good point in the feature space. Moreover, parameter tuning often calls for the use of cross-validation to improve the generalization properties of the given machine learning techniques. And it increases the number of trainings and evaluations even more. Thus, the whole parameter tuning process may require large amount of computational resources, if this aspect is not taken into account.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 61–66, 2013.

The use of surrogate models aims at lowering the number of objective function evaluations directly by the means of modeling of the objective function. The most straightforward way to apply the surrogate modeling for the problem of parameter tuning would be to model the classification error of the classifier based on its parameters. However, as we have already discussed (the first challenge above), most of the parameter settings yield similar values. This can be problem for some of the surrogate models.

In this paper we describe our multi-objectivization approach for tuning of the parameters of Radial Basis Function (RBF) Networks and Multilayer Perceptron (MLP) [6] networks with the goal to provide settings which minimize the classification error. The approach is compared to the performance of single-objective evolutionary algorithm. Moreover, surrogate versions of both the single-objective and multi-objective algorithms are discussed. Some of the ideas (namely the multi-objectivization) were already presented as an abstract [12].

The rest of the paper is organized as follows: in the next section we provide an overview of existing approaches from the literature. Section 3 describes our multi-objectivization approach. In Section 4 we present the experiment setup and the data sets we used for testing as well as the results of the different methods and, finally, Section 5 concludes the paper and provides some ideas for future research.

2 Related Work

The first attempts at parameter tuning were specifically designed for a given type of classifier, for example there are several algorithms to optimize the parameters of SVM [5, 8].

Kohen et al. [9] proposed a framework for Tuned Data Mining. The framework contains both feature selection and parameter tuning. The parameter tuning is done in the SPOT [2] framework, which also uses surrogate modeling.

Bergstra et al. [3] have used parameter tuning to enhance the performance of Deep Belief Networks (DBN). They used surrogate assisted evolutionary algorithm and showed that it outperforms both manual setting of parameters and random search on two data sets from the image recognition domain. Reif et al. [13] used several best performing settings on similar datasets as the individuals in the initial population to enhance the performance of evolutionary algorithms for parameter tuning.

Multi-objective optimization has also been used for regularization. Adding a regularizing term can be considered as another objective [7].

Another possibility to use multi-objective optimization for parameter tuning is so called multi-objectivization. Brockhoff et al. [4] showed that adding another objective may lead to a better performance even if only one objective is important.

3 Multi-objectivization

Our goal is to provide good parameter settings for a given classifier. The quality of the settings is measured by the classification error of the classifier.

In order to improve the convergence rate of the algorithm, we use the idea of multi-objectivization. We add two more objectives whose values are not important for our task, but which are correlated to the error rate of the optimizer. Then, a multi-objective evolutionary algorithm is used to solve the multi-objective optimization problem.

The three objectives we optimize are: the classification error (minimization) – it expresses the percentage of incorrectly classified instances; the kappa statistic (maximization) – it expresses the inter-classifier (or classifier and training set in our case) agreement (i.e. the agreement two random classifiers would have if they had the same distributions of classes as the actual classifiers); the root mean square error (minimization) – it is traditionally used for regression tasks – for classification, the class indices are encoded as binary vectors with just one 1 on the position of the class index.

The error rate and kappa statistic are highly correlated, especially in cases where all the classes are represented by the same number of instances in the training set. This is also the reason to add the third objective, which may seem unrelated to classification.

Root mean square error (RMSE) is traditionally used as the objective which is minimized in regression tasks. As it is implemented here (i.e. the model is trained to predict the unary representation of the class label), the number does not tell much about the quality of the classifier itself. However, it is more sensitive to changes in the parameter settings. It is important to note that the only important measure is the classification error, and RMSE guides the optimization to an optima of the classification error – if the RMSE is zero the classification error is also zero.

A surrogate based multi-objective evolutionary algorithm with local search and pre-selection (LSPS-MOEA) [11], was used to optimize the hyper-parameters of two types of neural networks: RBF networks and MLP networks [6]. The types of models used during the surrogate modeling were chosen using the automated model selection scheme, we described earlier [10]. Due to space limitations, the reader is referred to those publication for details on the algorithm.

4 Experiments

We used the described approach to tune the parameters of RBF and MLP networks. We optimized four parameters of RBF networks: the number of clusters (integer between 2 and 10), the minimal width of the Gaussians (real number between 0.01 - 1.0), the ridge parameter for the logistic regression (real number between 10^{-9} and 10), and the maximum number of iterations for the logistic regression (integer between -1 and 50, -1 meaning “until convergence”).

We also optimized four parameters for MLP networks: (the learning rate – real number between 0.001 and 1.0), the momentum (real number between 0.0 and 0.9), the number of epochs to train through (integer between 1 and 10,000), and the percentual size of the validation set, which is used to terminate the training (integer between 0 and 100).

The other parameters of the multilayer perceptron were fixed. The neural network has one hidden layer. The number of neurons in this layer is the arithmetic average of

Table 1. Performance of the algorithms while optimizing the RBF network

	Simple EA			Surrogate EA		
	best	mean	worst	best	mean	worst
balance-scale	0.0512	0.0712	0.1056	0.0528	0.0739	0.1040
breast-w	0.0286	0.0308	0.0329	0.0300	0.0315	0.0329
car	0.0741	0.0779	0.0839	0.0723	0.0802	0.0938
haberman	0.2386	0.2474	0.2516	0.2320	0.2412	0.2484
Iris	0.0200	0.0300	0.0333	0.0267	0.0287	0.0333
	Multi-objective EA			LSPS-MOEA		
	best	mean	worst	best	mean	worst
balance-scale	**0.0464**	0.0498	0.0528	**0.0464**	**0.0494**	**0.0512**
breast-w	0.0286	0.0296	0.0315	**0.0272**	**0.0285**	**0.0286**
car	0.0712	0.0737	0.0787	**0.0700**	**0.0719**	**0.0723**
haberman	**0.2288**	**0.2395**	0.2516	0.2353	**0.2395**	**0.2451**
iris	**0.0133**	**0.0200**	**0.0267**	0.0200	0.0220	**0.0267**

the number of attributes and number of classes. The ranges for the number of clusters of the RBF network was set in a range which corresponds to our experience with this particular model. Other ranges contain all (meaningful) values for the given parameters.

The performance of the tuners was tested on five datasets from the UCI Machine Learning Repository [1]: balance-scale, breast-w, car, haberman, and iris.

To obtain the results, the tuners were given the computational budget of 300 objective function evaluations. One evaluation is a 10-fold cross-validation with the parameters given by the tuner on the respective training set. All evaluated individuals are saved in an archive and if the same individual is generated again in the same run it is not re-evaluated. The archive of evaluated individuals is also used during the training of the surrogate model in the case of surrogate evolutionary algorithm.

The single-objective evolutionary algorithm (both with and without surrogate modeling) uses population of 10 individuals, with one point crossover and Gaussian mutation with the standard deviation of 30% of the range of the particular parameter. It uses tournament selection and 10% elitism (one individual). The version with surrogate model uses Gaussian process regression as the surrogate model. The local search operator is applied with the probability of 0.25 and uses another evolutionary algorithm with the same population size and operators which is run for 10 generations (only the standard deviation of the Gaussian mutation is reduced to 10% of the range to promote exploitation). The surrogate operator is not used, if there are less than 20 evaluated individuals in the archive.

The non-surrogate multi-objective evolutionary algorithm uses again the same parameters, the only difference being in the selection phase, where it uses the NSGA-II selection based on dominance and crowding distance. The parameters of LSPS-MOEA match those of the surrogate evolutionary algorithm with the NSGA-II selection.

Generally, the results (see Tables 1 and 2) correspond to the discussion we presented earlier. It is difficult for the surrogate model to precisely approximate the classification error of the optimizer, thus the surrogate model does not improve the performance of the

Table 2. Performance of the algorithms while optimizing the MLP network

	Simple EA			Surrogate EA		
	best	mean	worst	best	Mean	worst
balance-scale	0.0752	0.0776	0.0816	0.0736	0.0787	0.0848
breast-w	0.0300	0.0316	**0.0329**	0.0300	0.0319	0.0343
haberman	0.2288	0.2337	0.2386	0.2320	0.2343	0.2418
Iris	**0.0200**	**0.0207**	**0.0267**	**0.0200**	0.0247	**0.0267**
	Multi-objective EA			LSPS-MOEA		
	Best	mean	worst	best	mean	worst
balance-scale	**0.0688**	0.0746	**0.0784**	**0.0688**	0.0747	0.0800
breast-w	**0.0286**	**0.0313**	**0.0329**	0.0315	0.0319	**0.0329**
haberman	0.2288	0.2330	0.2386	**0.2255**	**0.2314**	**0.2353**
Iris	**0.0200**	0.0213	**0.0267**	**0.0200**	**0.0207**	**0.0267**

optimizer. However, we can observe, that the results of the surrogate algorithm are often better with respect to the worst and average run – this indicates that the surrogate versions have more robust performance, thus providing better guarantee on the optimality of the found solution. This is a feature which may be of importance in practice, where running the experiment several times to get better results may be unacceptable or intractable. On the other hand, as can be seen from the table, the difference between the single-objective and multi-objective version of the optimizers is significant. The multi-objective optimizers perform better than their single-objective version in almost all the cases.

In the case of RBF networks, the worst performance of the surrogate multi-objective optimizer in three of the five cases at least matches the best performance of the single-objective optimizer. On the haberman dataset, the the worst performance of LSPS-MOEA is better than the average one of simple EA. Multi-objective EA is the best for the iris dataset, however, the difference is not large (0.067 accounts for only one wrongly classified sample).

In the case of MLP networks the differences among the methods are much lower than in the case of RBF networks. It may be the case that MLP networks are less sensitive to the particular settings of the parameters we tried to optimize. Table 2 also indicates that all the optimizers were able to find similar optima on some of the datasets, which may indicate these are the true optima given the fixed parameters.

5 Conclusions

We have shown that using multi-objectivization for the parameter tuning may be more useful than using surrogate modeling. It is caused mainly by the specific type of the fitness function to be modeled, which leads to poorly trained models. On the other hand, adding more objectives, which are not in fact directly important for the optimization task at hand, may improve the results.

We have also shown that surrogate modeling does not improve the results much, but it is able to provide better results in the average and worst cases, which implies

better robustness of the results – this may also be important in practice, where running each experiment several times to get good results is not acceptable.

In the future, this approach will be used in a meta-learning case: for a given classification task, the meta-learning system will recommend the type of classifier together with the region of interest of the parameters, and the multi-objectivization will be used to tune the parameters.

Acknowledgement. Martin Pilat has been supported by the Charles University Grant Agency project no. 345511 and Roman Neruda has been partially supported by The Ministry of Education of the Czech Republic under project no. LD13002.

References

1. Asuncion, D.N.A.: UCI machine learning repository (2007)
2. Bartz-Beielstein, T., Lasarczyk, C., Preuss, M.: Sequential parameter optimization. In: Congress on Evolutionary Computation, pp. 773–780. IEEE (2005)
3. Bergstra, J., Bardenet, R., Bengio, Y., Kegl, B.: Algorithms for hyper-parameter optimization. In: NIPS 2011, pp. 2546–2554 (2011)
4. Brockhoff, D., Friedrich, T., Hebbinghaus, N., Klein, C., Neumann, F., Zitzler, E.: On the effects of adding objectives to plateau functions. Trans. Evol. Comp. 13(3), 591–603 (2009)
5. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Mach. Learn. 46(1-3), 131–159 (2002)
6. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall (July 1998)
7. Jin, Y.: Multi-objective machine learning. SCI, vol. 16. Springer, Heidelberg (2006)
8. Kapp, M.N., Sabourin, R., Maupin, P.: A PSO-based framework for dynamic SVM model selection. In: GECCO 2009, pp. 1227–1234. ACM, New York (2009)
9. Konen, W., Koch, P., Flasch, O., Bartz-Beielstein, T., Friese, M., Naujoks, B.: Tuned data mining: a benchmark study on different tuners. In: Proceedings of GECCO 2011, pp. 1995–2002. ACM, New York (2011)
10. Pilát, M., Neruda, R.: Meta-learning and model selection in multi-objective evolutionary algorithms. In: ICMLA (1), pp. 433–438. IEEE (2012)
11. Pilát, M., Neruda, R.: A surrogate based multiobjective evolution strategy with different models for local search and pre-selection. In: ICTAI 2012, pp. 1–8. IEEE (2012)
12. Pilát, M., Neruda, R.: Multiobjectivization for classiffer parameter tuning. In: GECCO 2013 (Companion), pp. 1–2. ACM (accepted 2013)
13. Reif, M., Shafait, F., Dengel, A.: Meta-learning for evolutionary parameter optimization of classiffers. Mach. Learn. 87(3), 357–380 (2012)

Automated Model Selection and Parameter Estimation of Log-Normal Mixtures via BYY Harmony Learning

Yifan Zhou, Zhijie Ren, and Jinwen Ma*

Department of Information Science, School of Mathematical Sciences and LMAM,Peking University, Beijing, 100871, China
jwma@math.pku.edu.cn

Abstract. Bayesian Ying-Yang (BYY) harmony learning system is a newly developed framework for statistical learning. Via the BYY harmony leaning on finite mixtures, model selection can be made automatically during parameter learning. In this paper, this automated model selection learning mechanism is extended to logarithmic normal (log-normal) mixtures. Actually, an adaptive gradient BYY harmony learning algorithm is proposed for log-normal mixtures. It is demonstrated by the experiments that the proposed BYY harmony learning algorithm not only automatically determines the number of actual log-normal distributions in the sample dataset, but also leads to a satisfactory estimation of the parameters in the original log-normal mixture.

Keywords: Bayesian Ying-Yang (BYY) harmony learning, Automated model selection, Parameter estimation, Logarithmic normal (log-normal) mixture.

1 Introduction

In data analysis and information processing, the finite mixture model is a common and powerful statistical tool [1]. When the number k of components in the mixture is unknown, an appropriate value of k must be selected or tested before or with the estimation of other parameters in the mixture model, which is a rather difficult task. Particularly, as the number of components is just a scale of the finite mixture model, its selection is usually referred to as model selection. Here, our interest focuses on the compound modeling of finite mixture for both parameter estimation or learning and model selection only with a given set of sample data.

For solving this compound mixture modeling problem, the traditional method is to choose the optimal number k^* of components via one of information, coding and statistical selection criteria such as Akaike's information criterion (AIC) [2] and Bayesian inference criterion (BIC) [3]. However, the process of evaluating a criterion incurs a large computational cost since the entire parameter estimation needs to be repeated at different value of k. Alternatively, under the framework of the EM (Expectation Maximization) algorithm with the MML (Minimum Message Length) criterion, Figueiredo and Jain [4] proposed an unsupervised learning algorithm with adaptive model selection where the extra components are discarded as soon as their mixing proportions become small enough during the learning process.

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 67–72, 2013.

From the point of view of statistical learning, Bayesian Ying-Yang (BYY) harmony learning system and theory [5] have provided a new theoretical framework for solving this compound mixture modeling problem. Actually, it has been solved on Gaussian mixtures in [6-8] through the maximization of a harmony function which leads to the parameter learning with automated model selection. As for the cases of non-Gaussian mixtures, the BYY harmony learning algorithms have been already established for Poisson and Weibull mixtures [9-10]. In the current paper, we extend this BYY harmony learning mechanism to logarithmic normal (log-normal) mixtures, which are also a typical class of non-Gaussian mixtures. Under a BI-directional architecture (BI-architecture) of the BYY harmony learning system for log-normal mixtures, an adaptive gradient learning algorithm for maximizing the harmony function is proposed to implement the parameter learning of log-normal mixture with automated model selection. It is demonstrated by the experiments that the proposed BYY harmony learning algorithm not only automatically determines the number of actual log-normal distributions in the dataset, but also leads to a satisfactory estimation of the parameters in the original or true log-normal mixture.

The rest of this paper is organized as follows. We begin to introduce the Log-normal mixture model and then propose the BYY harmony learning algorithm for log-normal mixtures in Section 2. Section 3 presents the experimental results and comparisons. Finally, we make a brief conclusion in Section 4.

2 Methods

Mathematically, the log-normal density takes the following form:

$$p(\mathbf{x}\,|\,\theta) = p(\mathbf{x}\,|\,\mathbf{m},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}}\,|\boldsymbol{\Sigma}|^{\frac{1}{2}}\prod_{i=1}^{n} x_i}\exp\{-\frac{1}{2}(\ln\mathbf{x}-\mathbf{m})^T\boldsymbol{\Sigma}^{-1}(\ln\mathbf{x}-\mathbf{m})\}, \tag{1}$$

where $\mathbf{m}$ is the mean vector, $\boldsymbol{\Sigma}$ is the covariance matrix, $\mathbf{X} = [x_1,\cdots,x_n]^T (x_i > 0)$ and $\ln\mathbf{x} = [\ln x_1, \ln x_2, \ldots \ln x_n]^T$. As for the log-normal mixture model, it is just a special type of the finite mixture model $q(\mathbf{x}\,|\,\boldsymbol{\Theta}_k) = \sum_{j=1}^{k}\alpha_j q(\mathbf{x}\,|\,\boldsymbol{\theta}_j)$ with $q(\mathbf{x}\,|\,\boldsymbol{\theta}_j)$ being the log-normal densities, where k is the number of components, and $\alpha_j \geq 0$ are mixing proportions of the components under the constraint that $\sum_{j=1}^{k}\alpha_j = 1$. For clarity, we let $\boldsymbol{\Theta}_k = \{\alpha_j, \boldsymbol{\theta}_j\}_{j=1}^{k}$ be the set of all parameters in the mixture model.

For the general finite mixture modeling, a BI-architecture of the BYY learning system has been already established such that its BYY harmony learning is equivalent to the parameter learning with automated model selection on finite mixtures [6-10]. Actually, given a dataset $\mathbf{D}_x = \{\mathbf{x}_t\}_{t=1}^{N}$ generated from a certain finite mixture model, the learning task on this architecture is to maximize the following harmony function on the finite mixture model with the parameter set $\boldsymbol{\Theta}_k = \{\alpha_j, \boldsymbol{\theta}_j\}_{j=1}^{k}$ given by

$$J(\boldsymbol{\Theta}_k) = \frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{k}\frac{\alpha_j q(\mathbf{x}_t\,|\,\boldsymbol{\theta}_j)}{\sum_{i=1}^{k}\alpha_i q(\mathbf{x}_t\,|\,\boldsymbol{\theta}_i)}\ln[\alpha_j q(\mathbf{x}_t\,|\,\boldsymbol{\theta}_j)]. \tag{2}$$

Letting $\mathbf{U}_j(x) = \alpha_j q(\mathbf{x} \mid \boldsymbol{\theta}_j)$, $p(l \mid x) = \mathbf{U}_l(x) / \sum_{j=1}^{k} \mathbf{U}_j(x)$, $\alpha_j = \exp(\beta_j) \Big/ \sum_{i=1}^{k} \exp(\beta_i)$, where $-\infty < \beta_j < +\infty$, $\lambda_i(t) = 1 - \sum_{l=1}^{k} (p(l \mid \mathbf{x}_t) - \delta_{il}) \ln \mathbf{U}_l(\mathbf{x}_t)$, we get the following adaptive gradient learning rule of β_j and $\boldsymbol{\theta}_j$:

$$\Delta\beta_j = \frac{\eta}{q(\mathbf{x}_t \mid \boldsymbol{\Theta}_k)} \sum_{i=1}^{k} \lambda_i(t)(\delta_{ij} - \alpha_j)\mathbf{U}_i(\mathbf{x}_t), \tag{3}$$

$$\Delta\theta_j = \frac{\eta\lambda_j(t)\alpha_j}{q(\mathbf{x}_t \mid \boldsymbol{\Theta}_k)} \frac{\partial q(\mathbf{x}_t \mid \theta_j)}{\partial \theta_j} = \eta q(j \mid \mathbf{x}_t)\lambda_j(t) \frac{\partial \ln q(\mathbf{x}_t \mid \theta_j)}{\partial \theta_j}, \tag{4}$$

where $\eta > 0$ is the learning rate which starts from a reasonable initial value, and decreases gradually to zero.

For the log-normal mixture model, we have the specific partial derivatives of the log-normal density function with respect to $\mathbf{m}_j$ and $\boldsymbol{\Sigma}_j$ as follows:

$$\frac{\partial p(\mathbf{x}_t \mid \mathbf{m}_j, \boldsymbol{\Sigma}_j)}{\partial \mathbf{m}_j} = p(\mathbf{x}_t \mid \mathbf{m}_j, \boldsymbol{\Sigma}_j)\boldsymbol{\Sigma}_j^{-1}(\ln \mathbf{x}_t - \mathbf{m}_j), \tag{5}$$

$$\frac{\partial p(\mathbf{x}_t \mid \mathbf{m}_j, \boldsymbol{\Sigma}_j)}{\partial \boldsymbol{\Sigma}_j} = \frac{1}{2} p(\mathbf{x}_t \mid \mathbf{m}_j, \boldsymbol{\Sigma}_j)[\boldsymbol{\Sigma}_j^{-1}(\ln \mathbf{x}_t - \mathbf{m}_j)(\ln \mathbf{x}_t - \mathbf{m}_j)^T - \mathbf{I}_n]\boldsymbol{\Sigma}_j^{-1}, \tag{6}$$

where $\mathbf{I}_n$ is an n-dimensional identity matrix. Thus, the adaptive gradient learning rule of $\mathbf{m}_j$ can be given as follows:

$$\Delta\mathbf{m}_j = \eta p(j \mid \mathbf{x}_t)\lambda_j(t)\boldsymbol{\Sigma}_j^{-1}(\ln \mathbf{x}_t - \mathbf{m}_j). \tag{7}$$

On the other hand, we utilize the decomposition technique suggested in [7]: $\boldsymbol{\Sigma}_j = \mathbf{B}_j\mathbf{B}_j^T$ to guarantee $\boldsymbol{\Sigma}_j$ to be always positive definite after each iteration, where $\mathbf{B}_j$ is a nonsingular square matrix. Via this decomposition, we can get the following adaptive gradient learning rule of $\mathbf{B}_j$:

$$\Delta vec\mathbf{B}_j = \frac{\eta}{2} p(j \mid \mathbf{x}_t)\lambda_j(t) vec\{[\boldsymbol{\Sigma}_j^{-1}(\ln \mathbf{x}_t - \mathbf{m}_j)(\ln \mathbf{x}_t - \mathbf{m}_j)^T - \mathbf{I}_n]\boldsymbol{\Sigma}_j^{-1}\} \frac{\partial(\mathbf{B}_j\mathbf{B}_j^T)}{\partial \mathbf{B}_j}, \tag{8}$$

where *vec*(**M**) denotes the vector obtained by stacking the column vectors of the matrix **M** and the detailed expression of $\frac{\partial(\mathbf{B}_j\mathbf{B}_j^T)}{\partial \mathbf{B}_j}$ can be found in [7].

Summing up the results of Eqs (4), (8) & (9), we have established the adaptive gradient learning algorithm for maximizing the harmony function on log-normal mixtures. In each iteration with a sample point $\mathbf{x}_t$, $\beta_j, \mathbf{m}_j$ and $\mathbf{B}_j$ will be adaptively updated. Accordingly, α_j and $\boldsymbol{\Sigma}_j$ will be also updated. As long as the difference of the harmony functions at the two sequential steps is small enough, the algorithm will stop and output the current values of the parameters in the log-normal mixture. In our

implementation, we initialize the number of components k greater the true number k^*, then there are k-k^* extra components in the algorithm. In order to improve the convergence performance of the algorithm, we can delete a component directly if its mixing proportion is small enough. Moreover, we can also combine two components if they are similar enough.

3 Experimental Results

We test the efficiency of our proposed adaptive gradient BYY harmony learning algorithm on four 2-D synthetic datasets. Actually, these four datasets are generated from four typical log-normal mixtures with different structures and overlaps among the components. The true parameters of log-normal mixtures for these four datasets are listed in Table 1.

We implement the adaptive gradient learning algorithm 20 times on each of these four 2-D datasets. Particularly, k is selected in $[k^*,3k^*]$ and the initial mean vectors are randomly selected from the dataset. Actually, in each experiment, the actual log-normal densities or distributions as well as their mixing proportions are correctly detected, with the extra components being discarded or combined. That is, the correct model selection is made automatically during the parameter learning. Moreover, the parameters in the original log-normal mixture are estimated with a good accuracy, which can be seen from Table 2 where the average estimates of the true parameters over 20 simulation results for each 2-D synthetic datasets are listed.

We further compare our proposed adaptive gradient BYY harmony learning (referred to as AGL-BYY) algorithm with the MML-based unsupervised learning (referred to as UL-MML) algorithm [4] particularly for log-normal mixtures, which has been considered as a typical and powerful existing learning algorithm for the

Table 1. The true parameters of log-normal mixtures for four synthesized 2D datasets

Dataset	j	m_{j1}	m_{j2}	σ_{11}^{j}	σ_{12}^{j}	σ_{22}^{j}	α_j	N_j
S_1	1	2.50	0	0.50	0	0.50	0.25	400
N=1600	2	0	2.50	0.50	0	0.50	0.25	400
	3	-2.50	0	0.50	0	0.50	0.25	400
	4	0	-2.50	0.50	0	0.50	0.25	400
S_2	1	2.50	0	0.45	-0.25	0.55	0.34	544
N=1600	2	0	2.50	0.65	0.20	0.25	0.28	448
	3	-2.50	0	1	0.10	0.35	0.22	352
	4	0	-2.50	0.30	0.15	0.80	0.16	256
S_3	1	2.5	0	0.10	-0.20	1.25	0.50	600
N=1200	2	0	2.5	1.25	0.35	0.15	0.30	360
	3	-1	-1	1	-0.80	0.75	0.20	240
S_4	1	2.50	0	0.28	-0.20	0.32	0.34	68
N=200	2	0	2.50	0.34	0.20	0.22	0.28	56
	3	-2.50	0	0.50	0.04	0.12	0.22	44
	4	0	-2.50	0.10	0.05	0.50	0.16	32

Table 2. The average estimates of the true parameters of the log-normal mixture for each of the four 2-D synthetic datasets over 20 simulation results

Dataset	j	$\hat{m}_{j1}$	$\hat{m}_{j2}$	$\hat{\sigma}_{11}^{j}$	$\hat{\sigma}_{12}^{j}$	$\hat{\sigma}_{22}^{j}$	$\hat{\alpha}_j$
S1	1	2.5762	0.0030	0.3792	-0.0236	0.4470	0.2502
k=7	2	0.0754	2.4896	0.4406	-0.0096	0.5122	0.2509
	3	-2.5366	0.0785	0.4367	-0.0008	0.4579	0.2458
	4	-0.0384	-2.5483	0.5458	0.0251	0.4872	0.2532
S2	1	2.5153	-0.0352	0.4477	-0.2355	0.5499	0.3534
k=7	2	0.0205	2.5086	0.6472	0.1824	0.2740	0.2874
	3	-2.6078	-0.0164	0.9236	0.0972	0.3407	0.2043
	4	-0.0360	-2.4638	0.3273	0.0904	0.7360	0.1549
S3	1	2.4597	-0.0078	0.1016	-0.1834	1.2063	0.5188
k=6	2	0.1095	2.5347	1.4365	0.3953	0.1639	0.2923
	3	-0.9429	-1.0349	1.0872	-0.9368	0.9470	0.1889
S4	1	2.4745	0.0260	0.2680	-0.2289	0.3537	0.3412
k=7	2	-0.1685	2.4088	0.2632	0.1278	0.1481	0.2822
	3	-2.4986	-0.0467	0.3784	0.0539	0.1412	0.2182
	4	0.0438	-2.5770	0.0976	0.0129	0.4385	0.1584

finite mixture modeling with adaptive model selection ability in a similar way. To do so, we run these two learning algorithms 20 times on each of the dataset $S_1 \sim S_4$ and give their average running times and parameter estimation errors on the four datasets in Table 3, where Δm is the estimation error of mean vector $\mathbf{m}$, $\Delta\sigma$ is the estimation error of covariance matrix Σ and $\Delta\alpha$ is the estimation error of mixing proportions $\boldsymbol{\alpha}$. These experiments are carried out on a computer with an Intel Core 2 Quad CUP Q8300 2.50GHz (2 GB RAM).

Table 3. The comparisons of the AGL-BYY and UL-MML algorithms on parameter estimation accuracy and running time

Dataset	Algorithm	Δm	$\Delta\sigma$	$\Delta\alpha$	Running Time(s)
S1	AGL-BYY	0.02290	0.03268	0.00003	23.64145
(k=7)	UL-MML	0.01918	0.01432	0.00001	37.95860
S2	AGL-BYY	0.01586	0.02034	0.00049	22.07485
(k=7)	UL-MML	0.01052	0.01054	0.00009	48.68045
S3	AGL-BYY	0.01882	0.15237	0.00055	25.38055
(k=6)	UL-MML	0.01284	0.10084	0.00002	35.04370
S4	AGL-BYY	0.06774	0.17998	0.00025	8.8149
(k=7)	UL-MML	0.04581	0.04273	0.00005	2.94295

According to the simulation results given in Table 3 as well as the further experimental results, we have found the following facts: (1). As for model selection and parameter estimation, the convergence results of the two learning algorithms are

similar on $S_1 \sim S_4$. However, in some synthetic datasets of complicated structure or small number of samples, the AGL-BYY algorithm can determine the correct number of components, but the UL-MML algorithm often leads to a wrong number of components; (2). The convergence speed of the AGL-BYY algorithm is generally faster than that of the UL-MML algorithm. Therefore, we can consider that the proposed adaptive gradient BYY learning algorithm is more efficient than the MML-based unsupervised learning algorithm for log-normal mixtures.

4 Conclusion

We have extended the BYY harmony learning mechanism to the log-normal mixture modeling by constructing an adaptive gradient BYY harmony learning algorithm for log-normal mixtures. It is demonstrated by the experiments that our proposed adaptive gradient BYY harmony learning algorithm is able to make model selection automatically and also to get a satisfactory estimation of the parameters in the original Log-normal mixture to generate the sample data, and even outperforms the MML-based unsupervised learning algorithm for log-normal mixtures as well.

Acknowledgements. This work was supported by the Ph.D. Programs Foundation of Ministry of Education of China for grant 20100001110006.

References

1. Mclachlan, G.J., Peel, D.: Finite mixture Models. John Wiley & Sons, New York (2000)
2. Akaike, H.: A new look at the statistical model identification. IEEE Trans. on Automatic Control AC-19, 716–723 (1974)
3. Scharz, G.: Estimating the dimension of a model. The Annals of Statistics 6, 461–464 (1978)
4. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(3), 381–395 (2002)
5. Xu, L.: BYY harmony learning, structural RPCL, and topological self-organizing on mixture modes. Neural Networks 15, 1231–1237 (2002)
6. Ma, J., Wang, T., Xu, L.: A gradient BYY harmony learning rule on Gaussian mixture with automated model selection. Neurocomputing 56, 481–487 (2004)
7. Ma, J., Wang, L.: BYY harmony learning on finite mixture: adaptive gradient implementation and a floating RPCL mechanism. Neural Processing Letters 24, 19–40 (2006)
8. Ma, J., Liu, J.: The BYY annealing learning algorithm for Gaussian mixture with automated model selection. Pattern Recognition 40(7), 2029–2037 (2007)
9. Ma, J., Liu, J., Ren, Z.: Parameter estimation of Poisson mixture with automated model selection through BYY harmony learning. Pattern Recognition 42(11), 2659–2670 (2009)
10. Ren, Z., Ma, J.: BYY Harmony Learning on Weibull Mixture with Automated Model Selection. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.) ISNN 2008, Part I. LNCS, vol. 5263, pp. 589–599. Springer, Heidelberg (2008)

A Simple but Robust Complex Disease Classification Method Using Virtual Sample Template

Shu-Lin Wang, Yaping Fang, and Jianwen Fang*

Applied Bioinformatics Laboratory, the University of Kansas,
2034 Becker Drive, Lawrence, KS 66047, USA
jwfang@ku.edu

Abstract. With the advance of high throughput technologies, genomic or proteomic data are accumulated rapidly, demanding robust computational algorithms for large-scale biological data analysis and mining. In this work we propose a simple classification method based on virtual sample template (VST) and three distance measurements. Each VST corresponds to a subclass in training set. The label of a test sample is simply determined by measuring the similarity between the test sample and each VST using the three distance measurements. The test sample is assigned to the subclass of the VST with the minimum distance. Our experimental results indicate that the proposed method is robust in predicative performance. Compared with other common classification methods of complex disease, our method is simpler and often with improved classification performance.

Keywords: Gene expression profiles, autoantibody profiles, complex disease classification, virtual sample template, correlation method.

1 Introduction

Complex diseases are caused by a combination of genetic, environmental, and lifestyle factors. Examples include Alzheimer disease (AD), Parkinson disease (PD), diabetes mellitus, cancers, and many others. The accurate classification of these complex diseases at the molecular level is of great benefit to the personalized diagnosis, treatment and prognosis. To achieve this goal, various classification methods have been applied to mining genomic or proteomic data such as gene expression profiles obtained by DNA microarray and autoantibody profiles obtained by protein microarray technologies [1-3]. Because these types of data have the characteristics of high dimensionality and small sample size, many existing classification methods focus on the dimensionality reduction of these data in order to construct prediction models [4-5].

The dimensionality reduction methods can be divided into two categories: feature selection and feature extraction. Each category has its own advantages and disadvantages [6]. Usually, features selected using feature selection methods have biological meaning, which can provide insight into the underlying mechanism of

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 73–80, 2013.

these datasets. However, because the high throughput data are usually noisy and all practical feature selection algorithms are heuristic, different algorithm often result in different sets of features, which make it difficult to identify features relevant to the biological problems under study. For feature extraction such as linear discriminant analysis (LDA) [7], it is hard to explain the meaning of the extracted features. Furthermore, although some complex classification methods such as support vector machines (SVM) can obtain high predictive accuracy, it is difficult to interpret the biomedical meaning of these complex methods. Thus, it is necessary to design more effective and more biomedical methods to recognize complex disease type, which is also the requirement of clinical application.

2 Methods

2.1 Analysis Framework

We design a novel but simple classification method based on virtual sample template (VST), where each VST corresponds a subclass in training set. Redundant or irrelevant features often degrade the classification performance, therefore differentially expressed features (genes or proteins) are firstly selected using a feature filter before further analysis. Fig. 1 shows the analysis framework of VST-based classification, which consists of four crucial steps described as following.

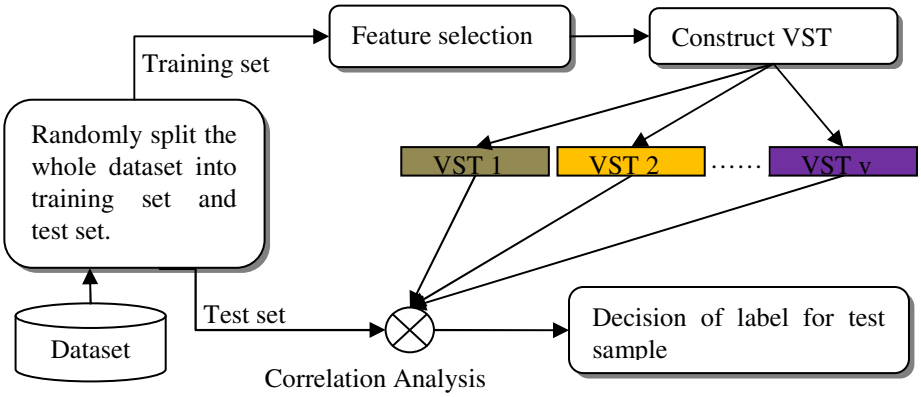


Fig. 1. The framework of our analysis method

Firstly, to avoid the bias of different ways on division of a whole dataset, we randomly split the whole dataset into two parts: a training set and a test set. Secondly, differentially expressed features are selected by adopting the feature filter technique that makes the following analysis only focusing on the differential portion. In our experiments the Kruskal-Wallis rank sum test (KWRST) method [8-9] is adopted to rank and select complex disease-related features. Thirdly, we construct a VST for each subclass in training set. This step plays an important role in predicting the label of test sample because each VST represents all samples in a subclass. Finally, we

measure the similarity between a test sample and each VST constructed in training set, and assign the test sample into the subclass with highest similarity. To obtain statistically sound results, all predictive accuracies are obtained by averaging the ones obtained by performing this procedure 200 times.

2.2 Classification Method Based on Virtual Sample Template

DNA microarray is composed of thousands of individual DNA sequences printed in a high density array on a glass microscope slide, while protein microarray is composed of thousands of proteins spotted in duplicate onto nitrocellulose-coated glass slides. Samples are generated under multiple conditions which may be a time series during a biological process or a collection of different tissue samples. The data collected by using the two techniques can be represented by a matrix in which each row denotes a sample and each column (called as feature) denotes a gene or protein. Let be a set of features and be a set of samples with subclasses , where each ,, denotes a subset including all samples belonging to the th subclass. The corresponding data matrix can be represented as , where denotes the number of samples, and denotes the number of features. The data matrix is composed of row vectors, and corresponds to , . Each sample , , represents a point in -dimensional space.

Firstly, the whole dataset is randomly split into the training set and the test set . We then construct each virtual sample template corresponding to a subclass ,. A simple mean method is designed to construct VST for each subclass in training set . Each VST, only belongs to in the training set , where . For each test sample , we adopt three distance measurement methods to compute the distance between the test sample and each VST , where and . Finally, we assign the label to the test sample .

(1) The Correlation distance is adopted as the distance measurement method, and it is computed by one minus the sample correlation between the test sample and each VST .

(1)

where and .

(2) For comparison, the Cosine distance is also adopted as the distance measurement method, and it is computed by the following formula.

(2)

(3) We also adopt Euclidean distance to measure the distance between the test sample and each VST , and it is computed by the formula (3).

(3)

3 Experiments

3.1 Six Complex Disease Datasets

We have downloaded six complex disease datasets to evaluate our method. They are Leukemia1 [10], Diffuse Large B-cell Lymphomas (DLBCL) [11], Leukemia2 [12], Small Round Blue Cell Tumor (SRBCT) [13], GCM [14], and GSE29676 [15] datasets (Table 1). The GSE29676 dataset includes 50 Alzheimer's disease and 29 Parkinson's disease samples as well as 40 non-demented control samples.

Table 1. The summary of six complex disease datasets

No.	Datasets	Platform	#Samples	#Features	#Subclasses
1	Leukemia1	Affy HGU95a	72	12,582	3
2	DLBCL	Affy HU6800	77	7,129	2
3	Leukemia2	Affy HU6800	72	7,129	3
4	SRBCT	cDNA	83	2,308	4
5	GCM	Affy HU6800	190	16,063	14
6	GSE29676	Invitrogen ProtoArray v5.0	119	9,480	3

3.2 Experimental Methods

The number of pre-selected features as well as the division of training set and test set can greatly affect the classification performance. Therefore, to obtain objective results the balance division method is adopted to divide each original dataset into balanced training set and test set and in the balanced training set the number of samples in each subclass is the same. We will demonstrate how the predictive performance varies with the different number of pre-selected features and the different divisions of training set and test set [5].

The VST-based classification method can be viewed as the 1-nearest neighbor (1NN) method with only one VST in each subclass. To show the superiority of the VST-based classification methods we will compare its results with the ones of classical KNN. According to different distance measurements adopted, we name the VST-based classification methods as VST-Correlation (adopt Correlation distance to measure the one between test sample and each VST), VST-Cosine (Cosine distance) and VST-Euclidean (Euclidean distance), respectively. And we call KNN-based classification methods as KNN-Correlation (adopt Correlation distance to measure the one between test sample and each training sample), KNN-Cosine (Cosine distance) and KNN-Euclidean (Euclidean distance), respectively. In our experiments we adopt 5NN (5-nearest neighbor) to classify test samples.

3.3 Results and Analysis

The resulting separability of all test samples (belonging to the same subclass) matching with each VST can be visualized using plots, from which we can visually determine which test sample is correctly or mistakenly classified. For example, the

SRBCT dataset contains 83 samples belonging to four subtypes, i.e., EWS, RMS, NB, and BL. In original SRBCT dataset we orderly select 5 samples per subtype as training set and the rest as test set. The four subplots corresponding to four subtypes are shown in Fig. 2, from which we can see that all test samples belonging to BL and NB subtypes are correctly classified, where the number of selected features is . The Y-axis denotes correlation degree, one minus the distance in formula (1), and the X-axis denotes the sequence number of test samples. For each test sample we can obtain four correlation degrees, and in each subplot we sequentially link all nodes that are the results of test samples matching with the same VST [5].

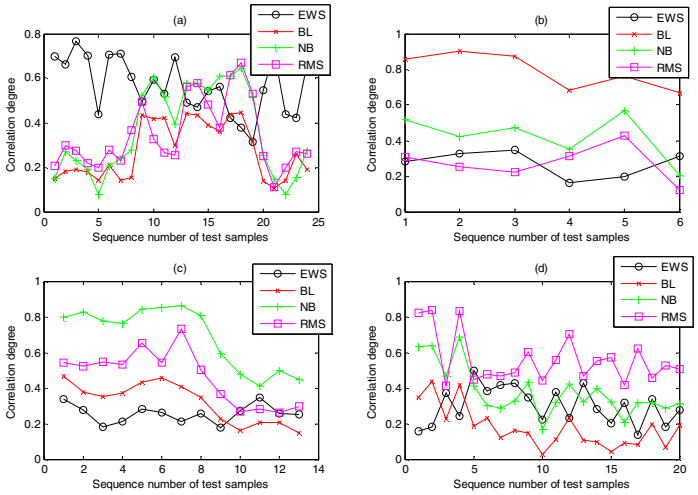


Fig. 2. The separability of all test samples in the SRBCT dataset. (a) The separability of all test samples belonging to EWS subtype matching with four VSTs, (b) BL, (c) NB, and (d) RMS.

The predictive performance of six methods varying with the different number of training samples on six complex disease datasets is shown in Fig. 3. For each dataset, features are pre-selected by using KWRST and the number of training samples per subclass is ranged from 5 to . Fig. 3 shows that all three VST-based methods slightly outperform the corresponding 5NN-based methods in predictive accuracies except on the GCM dataset. VST-based correlation and Cosine methods are the best two among the six methods. Furthermore, the predictive accuracies of the six methods generally increase with the increase of the number of training samples per subclass.

Fig. 4, in which features are pre-selected for each dataset and the number of training samples per subclass is fixed to 8, shows the prediction accuracies varying with different number of features on six datasets. Fig. 4 illustrates that the prediction accuracies of the six methods generally increase with the increase of the number of the pre-selected features; however, for the DLBCL and SRBCT datasets, too many features can lead to the decrease of predictive accuracies. Generally, VST-based methods also outperform KNN-based methods in predictive performance on five datasets except on the GCM dataset. The difference of predictive accuracies between VST-based correlation and KNN-based correlation method decreases with the increase of the selected genes.

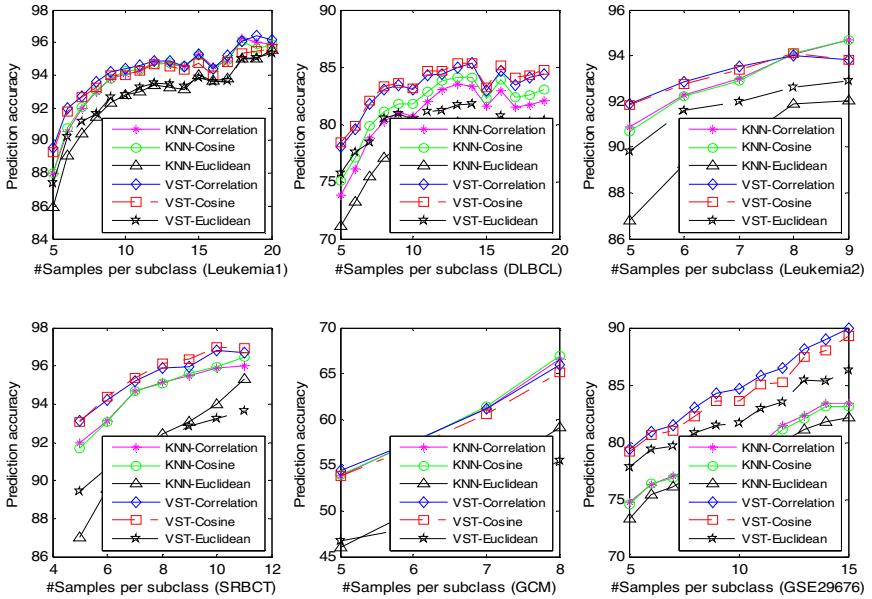


Fig. 3. The predictive accuracies of six methods varying with different number of training samples

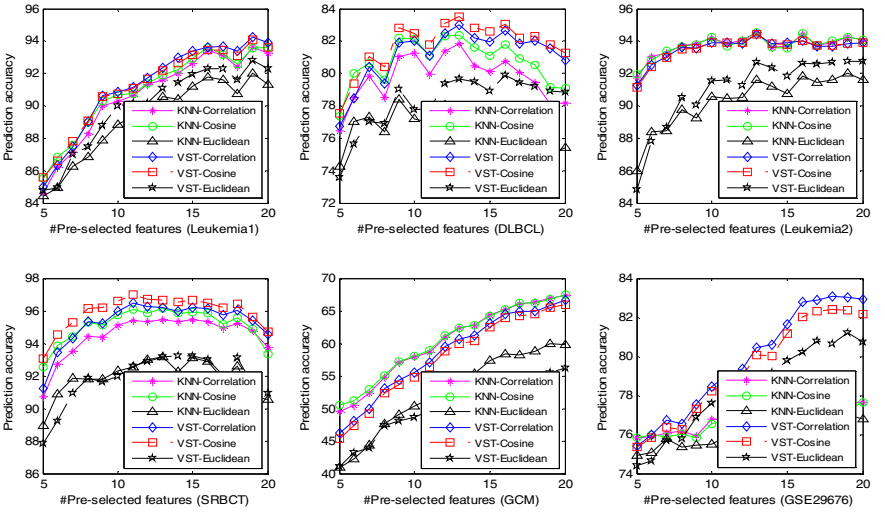


Fig. 4. The predictive accuracies of six methods varying with different number of features

Furthermore, Table 2 gives the accurate comparison of predictive accuracies of six methods on six datasets under the condition of that features are pre-selected and 8

samples in each subclass are divided into training set. Obviously, VST-based methods slightly outperform the corresponding KNN-based methods in predictive accuracies except on the GCM dataset.

Table 2. The comparison of prediction accuracies obtained by six methods

Datasets	VST-Correlation	VST-Cosine	VST-Euclidean	KNN-Correlation	KNN-Cosine	KNN-Euclidean
Leukemia1	93.57±3.46	93.23±3.76	91.66±4.16	92.99±3.90	93.08±3.92	91.42±4.47
DLBCL	82.98±7.53	83.34±7.19	80.55±8.10	80.24±8.84	81.08±8.37	77.08±9.77
Leukemia2	94.02±3.15	94.08±3.23	92.63±3.60	94.05±3.13	94.08±3.13	91.89±3.83
SRBCT	95.91±3.80	96.12±3.53	92.09±6.67	95.17±3.48	95.10±3.80	92.41±6.58
GCM	65.92±4.40	65.18±4.86	55.53±5.84	66.56±4.30	66.88±4.29	59.13±5.17
GSE29676	83.00±5.19	82.29±5.33	80.82±5.31	77.78±4.51	77.69±4.41	77.26±4.69

4 Conclusions

In this work we design a simple but robust method for complex disease classification in which each step can be intuitively interpreted from the viewpoint of biomedicine. Experiments on six public available complex disease datasets indicate that our method can achieve satisfactory performance, which is of great benefit to the clinical diagnosis and prognosis of complex disease. Another merit of our method is that our method does not require data normalization before classification, which might lead to the bias of results. Our future work will focus on the improvement of constructing VST for each subclass in training set because VST plays an important role in predicting the label of test sample.

Acknowledgments. This work is supported in part by the National Institutes of Health (NIH) Grant P01 AG12993 (PI: E. Michaelis) and the National Science Foundation of China (grant nos. 60973153, 61133010, 31071168, and 60873012).

References

1. Hanczar, B., Dougherty, E.R.: On the Comparison of Classifiers for Microarray Data. Current Bioinformatics 5, 29–39 (2010)
2. Wang, S., Li, X., Zhang, S.: Neighborhood rough set model based gene selection for multi-subtype tumor classification. In: Huang, D.-S., Wunsch II, D.C., Levine, D.S., Jo, K.-H. (eds.) ICIC 2008. LNCS, vol. 5226, pp. 146–158. Springer, Heidelberg (2008)
3. Wang, S.L., Li, X.L., Fang, J.W.: Finding minimum gene subsets with heuristic breadth-first search algorithm for robust tumor classification. Bmc Bioinformatics 13 (2012)
4. Nagele, E., Han, M., DeMarshall, C., Belinka, B., Nagele, R.: Diagnosis of Alzheimer's Disease Based on Disease-Specific Autoantibody Profiles in Human Sera. PLoS One 6 (2011)

5. Wang, S.L., Zhu, Y.H., Jia, W., Huang, D.S.: Robust Classification Method of Tumor Subtype by Using Correlation Filters. IEEE-ACM Transactions on Computational Biology and Bioinformatics 9, 580–591 (2012)
6. Asyali, M.H., Colak, D., Demirkaya, O., Inan, M.S.: Gene expression profile classification: A review. Current Bioinformatics 1, 55–73 (2006)
7. Sharma, A., Paliwal, K.K.: Cancer classification by gradient LDA technique using microarray gene expression data. Data Knowl. Eng. 66, 338–347 (2008)
8. Deng, L., Ma, J.W., Pei, J.: Rank sum method for related gene selection and its application to tumor diagnosis. Chinese Science Bulletin 49, 1652–1657 (2004)
9. Wang, S.-L., You, H.-Z., Lei, Y.-K., Li, X.-L.: Performance Comparison of Tumor Classification Based on Linear and Non-linear Dimensionality Reduction Methods. In: Huang, D.-S., Zhao, Z., Bevilacqua, V., Figueroa, J.C. (eds.) ICIC 2010. LNCS, vol. 6215, pp. 291–300. Springer, Heidelberg (2010)
10. Armstrong, S.A., Staunton, J.E., Silverman, L.B., Pieters, R., de Boer, M.L., Minden, M.D., Sallan, S.E., Lander, E.S., Golub, T.R., Korsmeyer, S.J.: MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. Nat. Genet. 30, 41–47 (2002)
11. Shipp, M.A., Ross, K.N., Tamayo, P., Weng, A.P., Kutok, J.L., Aguiar, R.C.T., Gaasenbeek, M., Angelo, M., Reich, M., Pinkus, G.S., Ray, T.S., Koval, M.A., Last, K.W., Norton, A., Lister, T.A., Mesirov, J., Neuberg, D.S., Lander, E.S., Aster, J.C., Golub, T.R.: Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. Nature Medicine 8, 68–74 (2002)
12. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 286, 531–537 (1999)
13. Khan, J., Wei, J.S., Ringner, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C.R., Peterson, C., Meltzer, P.S.: Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature Medicine 7, 673–679 (2001)
14. Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J.P., Poggio, T., Gerald, W., Loda, M., Lander, E.S., Golub, T.R.: Multiclass cancer diagnosis using tumor gene expression signatures. Proceedings of the National Academy of Sciences of the United States of America 98, 15149–15154 (2001)
15. Han, M., Nagele, E., DeMarshall, C., Acharya, N., Nagele, R.: Diagnosis of Parkinson's Disease Based on Disease-Specific Autoantibody Profiles in Human Sera. PLoS One 7 (2012)

Biweight Midcorrelation-Based Gene Differential Coexpression Analysis and Its Application to Type II Diabetes

Lin Yuan[1,2], Wen Sha[2], Zhan-Li Sun[2], and Chun-Hou Zheng[1,2,*]

[1] College of Information and Communication Technology,
Qufu Normal University, Rizhao, China
wfxueyuan@126.com
[2] College of Electrical Engineering and Automation, Anhui University, Hefei, China
zhengch99@126.com

Abstract. Differential coexpression analysis usually requires the definition of 'distance' or 'similarity' between measured datasets, the most common choices being Pearson correlation. However, Pearson correlation is sensitive to outliers. Biweight midcorrelation is considered to be a good alternative to Pearson correlation since it is more robust to outliers. In this paper, we introduce to use Biweight Midcorrelation to measure 'similarity' between gene expression profiles, and provide a new approach for gene differential coexpression analysis. The results show that the new approach performed better than three previously published differential coexpression analysis (DCEA) methods. We applied the new approach to a public available type 2 diabetes (T2D) expression dataset, and many additional discoveries can be found through our method.

Keywords: gene differential coexpression analysis, biweight midcorrelation, half-thresholding.

1 Introduction

DNA Microarray has been widely used as measurement tools in gene expression data analysis [1] [2] [3] [4]. Among the microarray data analysis methods, gene differential expression analysis is one of the most widely used types of analysis for disease research. However, genes and their protein products do not perform their functions in isolation [5] [6], but in cooperation.

Differential coexpression analysis, as a more comprehensive technique to the differential expression analysis, was raised to research gene regulatory networks and biological pathways of phenotypic changes. Differential coexpression genes are defined as genes whose correlated expression pattern differs between classes [7]. Methods for differential coexpression analysis of gene expression data have been extensively researched, and multiple algorithms have been developed and tested [8] [9] [10] [11]. Graeber [12] and Choi [5] both studied cancer from the perspective of

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 81–87, 2013.

differential coexpression. They found some genes were not be detected from the perspective of gene differential expression analysis. Carter [8] mined the molecular characteristics of the cell state through gene coexpression topology method.

In those gene differential coexpression analysis methods, the most common choice of similarity measurement is Pearson correlation [5] [11] [10] [13]. However, Pearson correlation is sensitive to outliers. Biweight midcorrelation (bicor) is considered to be a good alternative to Pearson correlation since it is more robust to outliers [14]. In this paper, based on combining Biweight Midcorrelation and half-thresholding, we proposed a new approach for gene differential coexpression analysis. The experiment results show that the approach performed better than three previously proposed differential coexpression analysis (DCEA) methods.

2 Methods

2.1 Biweight Midcorrelation

In order to define the biweight midcorrelation(bicor) [14] of two numeric vectors $x = (x_1, ..., x_m)$ and $y = (y_1, ..., y_m)$, we first defines u_i, v_i with i=1,…,m:

$$u_i = \frac{x_i - med(x)}{9mad(x)} \tag{1}$$

$$v_i = \frac{y_i - med(y)}{9mad(y)} \tag{2}$$

Where $med(x)$ is the median of x,and $mad(x)$ is the median absolute deviation of x.This lead us to the definition of weight w_i for x_i, which is,

$$w_i^{(x)} = (1-u_i^2)^2 I(1-|u_i|) \tag{3}$$

Where the indicator $I(1-|u_i|)$ take 1 if $1-|u_i| > 0$ and 0 otherwise. Given the weights, we can define biweight midcorrelation of x and y as:

$$bicor(x,y) = \frac{\sum_{i=1}^{m}(x_i - med(x))w_i^{(x)}(y_i - med(y))w_i^{(y)}}{\sqrt{\sum_{j=1}^{m}[(x_j - med(x))w_j^{(x)}]^2}\sqrt{\sum_{k=1}^{m}[(y_k - med(y))w_k^{(y)}]^2}} \tag{4}$$

2.2 'Half-Thresholding' Strategy in Constructing Gene Coexpression Networks

There are currently two accepted strategies, namely hard-thresholding and soft-thresholding. However, hard-thresholding dichotomizes the continuous correlation

values to be coexpression and non-coexpression. The soft-thresholding keeps all possible coexpression relationships and uses the power β (i.e. soft-threshold) to emphasize original high coexpression values and reduce the low coexpression values simultaneously. In our analysis, a pair of gene expression datasets under disease and normal conditions, are transformed to a pair of coexpression matrix. We let x_{ij} denote bicor coefficient between gene i and gene j under normal condition, and y_{ij} denote bicor coefficient under disease condition. The 'half-thresholding' strategy [13] keep coexpression value in both coexpression matrix if at least one of the two coexpression values exceeds the threshold.

2.3 The 'Biweight Midcorrelation and Half-Thresholding' method (BMHT)

In our method, the two datasets are encoded into a pair correlations matrix over all gene pairs. We then filter out non-informative correlation pairs using the half-thresholding strategy. This results in a subset of coexpression networks.

For gene i, the Biweight Midcorrelation coefficients between it and its n neighbors in the filtered set can be calculated from two vectors, i.e., $X=(x_{i1},x_{i2},\ldots,x_{in})$ and $Y=(y_{i1},y_{i2},\ldots,y_{in})$ for the two conditions.

$$dc_i(BMHT) = \sqrt{\frac{(x_{i1} - y_{i1})^2 + (x_{i2} - y_{i2})^2 + \ldots + (x_{in} - y_{in})^2}{n}} \tag{5}$$

Then we can use the *dc* values to rank genes. Finally, we perform a permutation test to evaluate the statistical significance of gene differential coexpression value. During the permutation test, we firstly randomly permute the disease and normal conditions of the samples 1000 times, then calculate new Biweight Midcorrelation coefficents, using 'half-thresholding' strategy based on the new values, finally calculate the *dc* statistics. The *p*-value for each gene can then be calculated.

3 Results and Discussion

3.1 Experiment Result on Simulate Datasets

In this experiment, we analyzed a pair of simulated datasets used a published study [15], which were generated based on two yeast signaling networks using SynTReN [16]. The simulate datasets consists of 20 genes, 50 samples in normal and disease conditions. MBP1_SWI6, PHO2, CLB5, TRP4, CLB6, FLO1, FLO10 were identified differential coexpression genes. We evaluated BMHT method in terms of its capability to discover the differential coexpression genes from the simulated datasets, and compared it with methods 'Log Ratio of Connection'(LRC), 'Average Specific Connection'(ASC), and 'Weighted Gene Coexpression Network Analysis'(WGCNA). We adopted the signed version of WGCNA and set the parameter β=12. The results are listed in Table 1. From Table 1 it can be seen that, the BMHT method can

detected all seven differential coexpression genes and ranked them at top, while the other three methods can not detect them.

Table 1. The twenty yeast genes involved in simulated dataset pair and the ranking of them by DCEA methods, signed WGCNA, ASC, and LRC separately

Gene	BMHT	Signed-WGCNA	ASC	LRC
MBP1_SWI6	1	7	1	8
PHO2	2	3	2	5
CLB5	3	14	3	18
TRP4	4	4	7	9
CLB6	5	16	4	19
FLO1	6	1	10	7
FLO10	7	2	6	3
CDC11	8	9	12	17
SWI4	9	5	5	16
ACE2	10	18	15	1
SWI4_SWI6	11	6	8	10
CDC10	12	10	13	12
ACT1	13	17	14	6
HTB1	14	8	11	15
LEU2	15	11	9	13
CTS1	16	12	17	14
SPT16	17	15	18	11
HO	18	13	16	2
CAF4	19	19	19	4
SNF6	20	20	20	20

Bold shown genes refers to the seven differential coexpression genes in the simulate datasets. We arranged the gene in accordance with the BMHT value.

3.2 Analyzing a Type 2 Diabetes(T2D) in Rats

In this section, we apply the new method to a pair of type 2 diabetes(T2D) in rats datasets (dataset pair T), which has been published in study [13]. Dataset pair T from dataset GSE3068 of Gene Expression Omnibus (GEO) database. Hui Yu et. al preprocessed dataset GSE3068. Dataset pair T includes 4765 genes in 10 T2D samples and 10 normal samples. After applied BMHT method to dataset pair T, we obtained 398 differential coexpression genes of 4765 genes (p-values cut-off 0.05, FDR<1.2%). The FDR is false discovery rate estimated from the p-value of biweight midcorrelation using Benjamini-Hochberg method [17]. In the differential coexpression genes, Rapgef4 [18] and Notch2 [19] are reported T2D-related genes. We listed all 25 differential coexpression genes with T2D relevance in table2. It is helpful for researchers excavate gene modules and disease genes, establish a disease-related gene clusters, further explore the pathogenesis of the disease and the biological function of the related-gene. The results are listed in table 2.

Table 2. Differential coexpression genes with existing evidence of T2D-relevance

Gene	BMHT value	Reported Relevance
Ucp2	0.7423	T2D-related
Rapgef4	0.7375	T2D-related
Nr5a1	0.7256	T2D-related
Inpp5d	0.7222	KEGG rno04910;T2D-related
Pparg	0.7068	T2D-related;T2D-associated
Igf1r	0.6885	KEGG rno04940
Tsc2	0.6706	KEGG rno04930
Jak3	0.6670	KEGG rno04940
Serpine1	0.6628	T2D-relaed
Lipe	0.6589	KEGGrno04910;T2D-related
C3	0.6581	T2D-related
Il6	0.6566	T2D-related
Foxo1	0.6550	KEGG rno04930
Flot2	0.6442	T2D-related
Prkab1	0.6432	KEGGrno04910;T2D-related
Pik3r1	0.6417	T2D-related
Gsk3a	0.6413	KEGG rno04930
Irf8	0.6391	KEGG rno04930
Tagln	0.6358	T2D-related
Slc2a1	0.6327	KEGG rno04930
Trf1	0.6324	KEGG rno04940
Cel	0.6322	T2D-related
Cckar	0.6254	T2D-related
Irs2	0.6220	KEGG rno04930
Notch2	0.6211	T2Dassociated;T2D-related

rno04940: type I diabetes mellitus; rno04930: type II diabetes mellitus; rno04910: insulin signaling pathway.

4 Conclusion

In this paper, we proposed a new approach for Differential coexpression analysis, which combine Biweight Midcorrelation and half-thresholding strategy. Biweight Midcorrelation is more robust for outliers and half-thresholding is an effective preprocess step of the proposed method. Experimental results on simulate datasets show that our method had better performance than three previsouly proposed methods. We apply the proposed BMHT method to real dataset designed for T2D study, and 398 differential coexpression genes were selected, which may be a useful resource for T2D study and explore the biological function of the related-gene. In the future, we will focus on how to introduce new measure to scale the similarity of gene pairs.

Acknowledgements. This work was supported by the National Science Foundation of China under Grant No. 61272339, the Natural Science Foundation of Anhui Province under Grant No. 1308085MF85, and the Key Project of Anhui Educational Committee, under Grant No. KJ2012A005.

References

1. Allison, D.B., Cui, X.Q., Page, G.P., Sabripour, M.: Microarray Data Analysis: from Disarray to Consolidation and Consensus. Nature Reviews Genetics 7, 55–65 (2006)
2. Baldi, P., Long, A.D.: A Bayesian Framework for The Analysis of Microarray Expression Data: Regularized t-test and Statistical Inferences of Gene Changes. Bioinformatics 17(6), 509–519 (2001)
3. Brown, M.P.S., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C.W., Furey, T.S., Ares Jr., M., Haussler, D.: Knowledge-based Analysis of Microarray Gene Expression Data by Using Support Vector Machines. Proc. Natl. Acad Sci. USA 97(1), 262–267 (2000)
4. Sturn, A., Quackenbush, J., Trajanoski, Z.: Genesis: Cluster Analysis of Microarray Data. Bioinformatics 18(1), 207–208 (2002)
5. Choi, J.K., Yu, U., Yoo, O.J., Kim, S.: Differential Coexpression Analysis Using Microarray Data and Its Application to Human Cancer. Bioinformatics 21(24), 4348–4355 (2005)
6. Rachlin, J., Cohen, D.D., Cantor, C., Kasif, S.: Biological Context Networks: A mosaic View of The Interactome. Mol. Syst. Biol. 2, 66 (2006)
7. Reverter, A., Ingham, A., Lehnert, S.A., Tan, S.H., Wang, Y., Ratnakumar, A., Dalrymple, B.P.: Simultaneous Identification of Differential Gene Expression and Connectivity in Inflammation, Adipogenesis and Cancer. Bioinformatics 22(19), 239–2404 (2006)
8. Carter, S.L., Brechbuhler, C.M., Griffin, M., Bond, A.T.: Gene Co-expression Network Topology Provides A Framework for Molecular Characterization of Eellular State. Bioinformatics 20(14), 2242–2250 (2004)
9. Mason, M.J., Fan, G., Plath, K., Zhou, Q., Horvath, S.: Signed Weighted Gene Co-expression Network Analysis of Transcriptional Regulation in Uurine Embryonic Stem Cells. BMC Genomics 10, 327 (2009)
10. Fuller, T.F., Ghazalpour, A., Aten, J.E., Drake, T.A., Lusis, A.J., Horvath, S.: Weighted Gene Coexpression Network Analysis Strategies Applied to Mouse Weight. Mammalian Genome 18(6-7), 463–472 (2007)
11. Freudenberg, J.M., Sivaganesan, S., Wagner, M., Medvedovic, M.: A Semi-parametric Bayesian Model for Unsupervised Differential Coexpression Analysis. BMC Bioinformatics 11, 234 (2010)
12. Graeber, T.G., Eisenberg, D.: Bioinformatic Identification for Potential Autocrine Signaling Loops in Cancers from Gene Expression Profiles. Nat. Genet. 29, 295–300 (2001)
13. Yu, H., Liu, B.H., Li, Y.Y.: Link-based Quantitative Methods to Identify Differentially Coexpressed Genes and Gene Pairs. BMC Bioinformatics 12, 315 (2011)
14. Wilcox, R.: Introduction to Robust Estimation and Hypothesis Testing. Academic Press, San Diego (1997)
15. Zhang, B., Li, H., Riggins, R.B., Zhan, M., Xuan, J., Zhang, Z., Hoffman, E.P., Clarke, R., Wang, Y.: Differential Dependency Network Analysis to Identify Condition Specific Topological Changes in Biological Networks. Bioinformatics 25(4), 526–532 (2009)

16. Bulcke, V.T., Leemput, V.K., Naudts, B., Remortel, P., Ma, H., Verschoren, A., Moor, D.B., Marchal, K.: SynTReN: A Generator of Synthetic Gene Expression Data for Design and Analysis of Structure Learning Algorithms. BMC Bioinformatics 7, 43 (2006)
17. Benjamini, Y., Hochberg, Y.: Controlling The False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society Series B 57, 289–300 (1995)
18. Scott, et al.: A Genome-wide Association Study of Type 2 Diabetes in Finns Detects Multiple Susceptibility Variants. Science 316(5829), 1341–1345 (2007)
19. Zeggini, et al.: Meta-analysis of Genome-wide Association Data and Large-scale Replication Identifies Additional Susceptibility Loci for Type 2 Diabetes. Nature Genetics 40, 638–645 (2008)

A Hybrid Gene Selection and Classification Approach for Microarray Data Based on Clustering and PSO

Shanxiu Yang, Fei Han, and Jian Guan

School of Computer Science and Telecommunication Engineering,
Jiangsu University, Zhenjiang, China
yangshanxiu_joy007@163.com, hanfei@ujs.edu.cn,
jianguan87@gmail.com

Abstract. In this paper, a novel hybrid approach based on clustering and particle swarm optimization (PSO) is proposed for gene selection and classification of microarray data. In this approach, PSO combining with clustering method are used to perform gene selection to reduce redundancy. Firstly, genes are partitioned into a certain number of clusters by using K-means, and then PSO is used to perform gene selection from the clustered genes. Because of its better generalization performance with much faster convergence rate than other learning algorithms for neural networks, extreme learning machine (ELM) is chosen to perform sample classification in the hybrid method. The proposed method selects less redundant interpretable genes as well as increases prediction accuracy. The efficiency and effectiveness of the proposed method is verified by extensive comparisons with other classic methods on some open microarray data.

Keywords: Gene selection, clustering, particle swarm optimization, extreme learning machine.

1 Introduction

Microarray data has a large number of genes and a relatively small sample size. Various methods have been applied to microarray data analysis, such as clustering algorithms, gene selection and classification methods. The methods for gene selection are broadly divided into three categories: filter, wrapper and embedded methods [1]. Clustering is one of the most common methods for discovering hidden structure in microarray data. Genes are partitioned into a few of clusters within each of which genes are expected to have similar functions. Bringing together marker genes of each cluster for gene selection can effectively reduce redundancy of microarray data. As for microarray data classification, many methods such as statistical techniques [2], support vector machine (SVM) [3], and back-propagation (BP) algorithm [4], have been effectively used.

Many evolutionary algorithms such as genetic algorithms (GA) [3] and particle swarm optimization (PSO) [5, 6] have been used for gene selection, because they have good performance on searching global minima [7]. Compared with GA, PSO has some attractive characteristics such as fast convergence rate, less parameter. As a

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 88–93, 2013.

learning algorithm for single-hidden layer feed-forward neural network (SLFNs), extreme learning machine (ELM) [8] tends to achieve the best generalization performance with thousands of times faster convergence rate than traditional learning algorithms such as backpropagation algorithm [9].

For reducing redundant genes and increasing classification accuracy, the KMeans-PSO-ELM approach is proposed in this paper, which combines clustering algorithm with PSO for gene selection, and then uses ELM for samples classification. First, K-means method is used to cluster preselected genes. Then PSO is employed to analyze each cluster select informative genes from each cluster. In the process of gene selection, ELM is used to evaluate the selected gene subsets. Finally ELM conducts classification on optimal selected gene subsets.

2 Related Methods

2.1 Particle Swarm Optimization

The PSO algorithm [5] was first developed and introduced as a stochastic optimization algorithm. It belongs to the category of swarm intelligence techniques which are inspired by the social behavior of flocking animals such as birds and fishes. PSO is a sociologically inspired population based optimization algorithm. Each particle is an individual, and the swarm consists of particles. In PSO, each position in the search space is a potential solution of the problem. At each iteration, each particle adjusts its velocity vector based on information about each particle previous best position (P_i) and the best position of its neighbors (P_g). For a D-dimensional search space the position of the i-th particle is represented as $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$. Each particle records its previous best position $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$ and a $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$ along each dimension. The best particle among all the particles in the population is represented by $P_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$. During each iteration, the particles are updated according to the following equation [6]:

$$v_{id} = w \times v_{id} + c_1 \times rand() \times (p_{id} - x_{id}) + c_2 \times rand() \times (p_{gd} - x_{id}) \tag{1}$$

$$x_{id} = x_{id} + v_{id} \quad 1 \le i \le n, 1 \le d \le D \tag{2}$$

where c_1, c_2 are positive constants; *rand()* is a random function in the range of [0,1]. w is an inertial weight. The parameter w can reduce gradually as the generation increases

2.2 Extreme Learning Machine

For N arbitrary distinct samples (x_i, t_i) ($i = 1, 2, \cdots, N.$), where $x_i = [x_{i1}, x_{i2}, \cdots, x_{in}]^T \in R^n$, $t_i = [t_{i1}, t_{i2}, \cdots, t_{im}]^T \in R^m$. A SLFN with H hidden neurons and activation function $g(\cdot)$ can approximate these N samples with zero error. This means that

$$Hwo = T \tag{3}$$

where $H(wh_1,\ldots,wh_H,b_1,\ldots,b_H,x_1,\ldots,x_N)$

$$= \begin{bmatrix} g(wh_1 \bullet x_1 + b_1) & \cdots & g(wh_H \bullet x_1 + b_H) \\ \vdots & \cdots & \vdots \\ g(wh_1 \bullet x_N + b_1) & \cdots & g(wh_H \bullet x_N + b_H) \end{bmatrix}_{N \times H}, \quad wo = \begin{bmatrix} wo_1^T \\ \vdots \\ wo_H^T \end{bmatrix}_{H \times m}, \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \tag{4}$$

The $wh_i = [wh_{i1}, wh_{i2}, \ldots, wh_{in}]^T$ is the weight vector connecting the *i*-th hidden neuron and the input neurons, the $wo_i = [wo_{i1}, wo_{i2}, \ldots, wo_{im}]^T$ is the weight vector connecting the *i*-th hidden neuron and the output neurons, and the b_i is the threshold of the *i*-th hidden neuron [9].

In the course of learning, first, the input weights and the hidden layer biases are arbitrarily chosen and need not be adjusted at all. Second, the smallest norm least-squares solution of the Eqn. (3) is obtained as follows:

$$wo = H^{+} T \tag{5}$$

From the above discussion, it can be found that the ELM has the minimum training error and smallest norm of weights. The smallest norm of weights tends to have the best generalization performance. Since the solution is obtained by an analytical method and all the parameters of SLFN need not be adjusted, ELM converges much faster than gradient-based algorithm.

3 The Proposed KMeans-PSO-ELM Approach

Since the microarray datasets and genes associated are very large, the searches for optimal gene subsets are very important. PSO algorithm has the advantage of finding global optimal solution. Through the clustering algorithm, we can study the impact of combination of genes on classification. Moreover, there is little possibility that ELM gets stuck in the local optimal solution. Hence, we combine K-means clustering algorithms with PSO for gene selection, and use ELM to perform samples classification with the selected genes. The proposed method is called KMeans-PSO-ELM. The detailed steps for KMeans-PSO-ELM are as follows.

Step1: Select 200~400 genes from training set of microarray data with Information Gain Ratio (IGR) to form an initial gene pool. The training set is further divided into training and validation sets.

Step2: Cluster all genes in the gene pool to predetermined number of groups by K-Means method. Record the index of genes of every cluster.

Step3: Initialize a population of particles with random positions and velocities in the search space. Each particle represents a feasible solution i.e. a gene subset. The dimension of a particle is equal to the number of clusters.

Step4: Evaluate the fitness value of each particle according to the fitness value, which is the classification accuracy obtained by ELM on validation set.

Step5: The search for the global optimal solution is made through dynamically updating the particles in the population. The velocity will be updated according to Eq. (1). Position update will be made using Eq. (2) by adding incremental change in position in each iteration. If the particles move out of boundary they will be reset to the boundary value to force them to search within the boundary.

Step6. Steps 4-5 are repeated until the termination condition is reached. The features returning the best validation accuracy eventually are selected as the final optimal genes.

Step7: With the selected genes, ELM performs samples classification on all data.

4 Experiment Results

To verify the effectiveness, the experiments on three open microdata (Leukemia, Colon and SRBCT) are conducted in this section. The Leukemia and Colon data are obtained at http://datam.i2r.a-star.edu.sg/datasets/krbd/, and the SRBCT data are obtained at (http://research.nhgri.nih.gov/microarray/Supplement/. The detailed description of three data are listed in Table 1.

Table 1. Three Microarray datasets

Dataset	Number of samples	Number of classes	Number of genes
Leukemia	72	2	7129
Colon	62	2	2000
SRBCT	83	4	2308

Table 2. The accuracy and serial number with different gene subsets on three datasets

Data	Selected genes	Training Accuracy mean(%) ± std	Testing Accuracy mean(%) ± std
Leukemia	4847,2642,4050	100±0.00	99.65±0.71
	2642,4050,2121	100±0.00	100±0.00
	2642,4050,2043	100±0.00	99.56±1.47
	2642,6510,1882	99.40±1.11	99.71±1.29
Colon	14,187,1993	93.17±0.50	93.53±0.98
	14,187,1993,1644	94.55±0.41	94.27±1.00
	14,187,1993,1644,1546	94.24±0.37	94.24±0.88
SRBCT	566,255,1194	91.08±0.66	90.73±1.68
	905,1645,255,509	98.16±0.41	96.51±1.33
	905,566,846,742,509	99.04±0. 44	99.10±0.87
	905,1645,846,255,742,509	100±0.00	100±0.00

Table 2 shows the results of five-fold cross validation (CV) with ELM on some selected gene subsets by the proposed method for 100 times. From Table 2, It is easy to obtain 100% classification accuracy on Leukemia and SRBCT data with the selected genes by the proposed approach. However, As for Colon data, the highest accuracy is 94.27%, and there is no increase of classification accuracy when the number of selected genes reaches 4 or more.

Table 3. Comparison of the relevant works on cancer classification with the proposed approach

Gene selection methods	Leukemia	Colon	SRBCT
KMeans-PSO-ELM	100(3)	93.53(3)	100(6)
GA-SVM[3]	100(6)	93.55(12)	-
MRMR-ELM[9]	100(11)	89.06(28)	
Gene ranking-Clustering[10]	100(8)	91.9(3)	-
KMeans-SNR-SVM[11]	100(5)	-	-
Evolutionary Algorithm[12]	100(9)	-	100(12)
MLP-ANN[13]	-	-	100(96)
SVM [14]	-	-	100(7)

Table 3 shows the test accuracies with some classical methods . As for Leukemia and SCRBCT data, all methods achieve 100% classification accuracy, but the proposed method obtains 100% classification accuracy with the least number of genes. For Colon data, although the proposed method does not obtain the highest accuracy, slightly lower than GA-SVM [3], it requires least number of genes.

5 Conclusions

To obtain high classification accuracy with low redundant genes, a hybrid method called KMeans-PSO-ELM was proposed in this paper. In the new method, PSO combining with KMeans clustering method is used to perform gene selection, and ELM is used to perform samples classification. The experiment results on three open microarray data verified that the KMeans-PSO-ELM method nearly obtained the highest classification with the least number of genes. Furture work will include improving the proposed method to apply to more complex microarray data.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Nos.61271385, 60702056) and the Initial Foundation of Science Research of Jiangsu University (No.07JDG033).

References

1. Saeys, Y., Inza, I., Larranaga, P.: A Review of Feature Selection Techniques in Bioinformatics. Bioinformatics 23(19), 2507 (2007)
2. Logsdon, B.A., Hoffman, G.E., Mezey, J.G.: Mouse Obesity Network Reconstruction with A Variational Bayes Algorithm to Employ Aggressive False Positive Control. BMC Bioinformatics 13, 53 (2012)

3. Peng, S., Xu, Q., Ling, X.B., et al.: Molecular Classification of Cancer Types from Microarray Data Using the Combination of Genetic Algorithms and Support Vector Machines. FEBS Letter 555(2), 358–362 (2003)
4. Lee, C.-M., Ko, C.-N.: Time Series Prediction Using RBF Neural Networks with A Nonlinear Time-varying Evolution PSO Algorithm. Neurocomputing 73(1-3), 449–460 (2009)
5. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
6. Shi, Y., Eberhary, R.C.: A Modified Particle Swarm Optimizer. Computational Intelligence 6, 69–73 (1998)
7. Han, F., Ling, Q.H., Huang, D.S.: An Improved Approximation Approach Incorporating Particle Swarm Optimization and A Priori Information into Neural Networks. Neural Computing & Applications 19(2), 255–261 (2010)
8. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme Learning Machine: Theory and Applications. Neurocomputing 70, 489–501 (2006)
9. Huang, G.B., Ding, X., Zhou, H.: Optimization Method Based Extreme Learning Machine for Classification. Neurocomputing 74, 155–163 (2010)
10. Wang, Y., Makedon, F., Ford, J.C., et al.: A Hybrid Approach for Selecting Marker Genes for Phenotype Classification Using Microarray Gene Expression Data. Bioinformatics 21(8), 1530–1537 (2005)
11. Mishra, D., Sahu, B.: Feature Selection for Cancer Classification: A Signal-to-noise Ratio Approach. International Journal of Scientific & Engineering Research 2(4) (2011)
12. Deutsch, J.M.: Evolutionary Algorithms for Finding Optimal Gene Sets in Microarray Prediction. Bioinformatics 19, 45–52 (2003)
13. Khan, J., Wei, J.B., et al.: Classication and Diagnostic Prediction of Cancers Using Gene Expression Proling and Articial Neural Networks. Nature Medicine 7(6), 673–679 (2001)
14. Chu, F., Wang, L.: Applications of Support Vector Machines to Cancer Classification with Microarray Data. Int. J. Neural Syst. 15(6), 475–484 (2005)

Manifold Learner Ensemble

Peng Zhang[1], Chunbo Fan[1], Yuanyuan Ren[2], and Nina Zhang[1]

[1] Data Center, National Disaster Reduction Center of China
Beijing, 100124, P.R. China
{zhangpeng,fanchunbo,zhangnina}@ndrcc.gov.cn
[2] Institute of Microbiology, Chinese Academy of Sciences
Beijing, 100101, P.R. China
renyy@im.ac.cn

Abstract. Manifold learning is proved to be an efficient tool for nonlinear dimensionality reduction. Various local and global learners have been proposed to successfully extract intrinsic geometry underlying high-dimensional data cloud. However, there is no work considering the ensemble of local and global manifold learners to promote learning results, where such strategy has achieved great success in classification. In this paper, we propose a manifold learner ensemble method (MLEN) for the first time. MLEN consists of a local manifold learner and a global one. By fusing the extracted local and global geometry, MLEN outperforms each of its components and outputs an overall and superior embedding. Experimental results on both synthetic and image manifolds validate the effectiveness of the proposed method.

Keywords: manifold learning, ensemble learning, dimensionality reduction, feature extraction.

1 Introduction

Manifold learning, as a promising tool for nonlinear dimensionality reduction, has been a hot research topic in computer science since proposed. By preserving local or global geometry, manifold learning methods can effectively learn meaningful and low-dimensional embeddings from manifold-modeled and high-dimensional data. Up to now, a large variety of manifold learning methods have been proposed and they can be cast into two categories, local and global methods, according to the geometry that is preserved. Local methods include Locally Linear Embedding (LLE) [7], Laplacian Eigenmap (LE) [3], and Local Tangent Space Alignment (LTSA) [11], and global ones include ISOMAP [8], Maximum Variance Unfolding (MVU) [9], and Riemannian Manifold Learning (RML) [6], to name just a few.

Despite the success of manifold learning methods in theory and application, a key issue with current approaches is that only one kind of geometric characteristic, local or global, is preserved in the learned embedding. As a consequence, the intrinsic geometry underlying high-dimensional data may not be fully exploited. Few works consider the ensemble of local and global manifold learners to simultaneously encode both local and global intrinsic geometry and get superior learning results.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 94–99, 2013.

However, such strategy has been widely and successfully used in classification, such as bagging and boosting [2], and it is proved to be an economic yet efficient way to greatly promote the overall performance [5].

In this paper, we proposed a manifold learner ensemble method, named as MLEN, which can preserve both local and global geometry of a data manifold. As far as we know, this is the first time that an ensemble of manifold learners is proposed. MLEN includes a local learner and a global learner, whose embeddings preserve local tangential coordinates and global geodesic distances, respectively. Then by fusing the learned embeddings through an ensemble learning process, MLEN can outperform any of its component learners and output an overall embedding. Besides, MLEN has a simple structure and low computational complexity, hence it is easy to implement. To validate the effectiveness of the proposed method, experiments on benchmark data sets and high-dimensional image manifold are conducted. Experimental results demonstrate that MLEN can efficiently encode both local and global geometric information in the learned embedding.

The rest parts of the paper are organized as follows. Section 2 describes the details of the proposed MLEN method. Section 3 presents experimental results on benchmark data sets. Section 4 gives some concluding remarks.

2 Manifold Learner Ensemble

2.1 Structure of the Ensemble

Current local and global manifold learning methods have their own shortcomings. On one hand, local ones normalized their embeddings, hence each coordinate is scaled separately and global geodesic distances are no longer preserved. On the other hand, global ones preserve approximated geodesic distances, and then local geometry may not be learned so accurately.

Motivated by ensemble of classifiers, an intuitive solution to address the above issue is to design an ensemble of manifold learners. In the ensemble, local and global manifold learning methods serve as components and learn data manifold separately. Then their learned embeddings are fused such that both local and global geometry are preserved in the final embedding. In this paper, we take two representative manifold learning methods, LTSA and ISOMAP, as components of the ensemble. A straightforward fusing approach is to match the embedding learned by LTSA to that of ISOMAP such that each coordinate is rescaled to its original value. By doing this, not only local topology defined by tangential coordinates still holds, but also geodesic distances are well approximated by global shape recovery.

2.2 Ensemble Learning Process

Formally, let $X = \{\underline{x}_1, x_2, \ldots, x_N\} \subset R^D$ and $Y = \{y_1, y_2, \ldots, y_N\} \subset R^d$ be high-dimensional training data and low-dimensional embedding, respectively. Let Y_L and Y_G denote the matrices of embeddings learned by LTSA and ISOMAP, respectively.

As described in Sect. 2.1, we aim to find an optimal rotation plus separate scaling along each coordinate such that Y_L best matches Y_G.

Mathematically, the aforementioned transformation can be expressed as

$$Y_G = RDY_L, \tag{1}$$

where R is an orthogonal matrix, that is, $R^TR = I_d$, and D is a d by d diagonal matrix with nonzero diagonal entries. Here we assume that both Y_L and Y_G are centered at the origin. Otherwise, they are preprocessed with mean removal.

Then the optimal $R^*$and D^* can be computed by solving the following optimization problem

$$\begin{aligned} \min_{R,D} \quad & \left\| Y_G - RDY_L \right\|_{\mathrm{F}} \\ \text{s.t.} \quad & R^{\mathrm{T}}R = I_d \end{aligned} . \tag{2}$$

By expanding the objective function, it can be proved that the optimization problem is convex over D. Then by Lagrange method, the optimal solution D^* is given by

$$D^* = \left(\delta\left(Y_L Y_L^T\right)\right)^{-1} \delta\left(R^T Y_G Y_L^T\right) , \tag{3}$$

where the δ operator transforms a square matrix into a diagonal one by keeping only its diagonal entries. By substituting (3) into (2), the optimization problem now reads

$$\begin{aligned} \max_{R} \quad & \mathrm{tr}\left(\left(R^T M\right) \bullet \left(R^T M\right)\right) \\ \text{s.t.} \quad & R^T R = I_d \end{aligned} , \tag{4}$$

where $M = Y_G Y_L^T \delta\left(Y_L Y_L^T\right)^{-1/2}$ and "•" stands for the entry-wise Hadamard product over matrices.

The above optimization problem does not admit a closed-form solution. However, it can be solved through gradient descent method over Stiefel manifold [1]. In our previous work [10], we proposed an efficient algorithm, named as ASIM, to address this issue. We do not present this algorithm here due to space limit. Interested readers can refer to [10] for more details. Once the optimal rotation and scaling matrices are computed, the final ensemble embedding Y is simply given by $Y = R^* D^* Y_L$.

MLEN has four parameters: k_l and k_g, the number of nearest neighbors for LTSA and ISOMAP, respectively; step length α and tolerance ε in ASIM. The computational complexity of MLEN equals to that of its component, which is at most $O(N^3)$.

3 Experimental Results

In this section, we conduct three experiments to validate the performance of MLEN. We first apply MLEN to learn two benchmark surfaces embedded in $\boldsymbol{R}^3$, namely the SwissRoll and SCurve manifolds. Then we test MLEN on a high-dimensional image manifold with controllable intrinsic degrees of freedom. In each experiment,

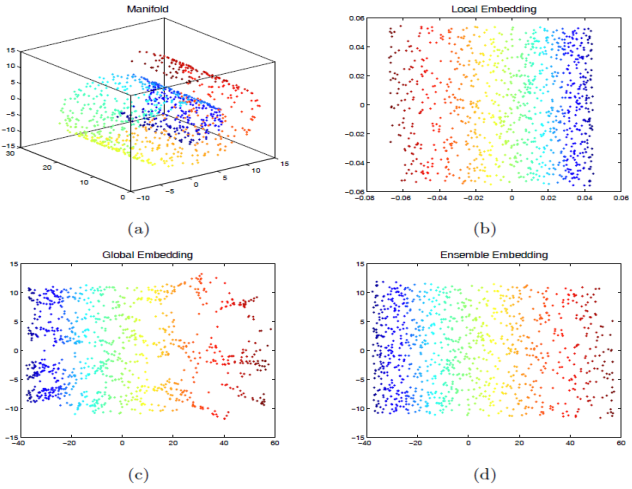


Fig. 1. Experimental results on the `SwissRoll` manifold. (a) The manifold. (b)-(d) Learning results by LTSA, ISOMAP, and MLEN, respectively.

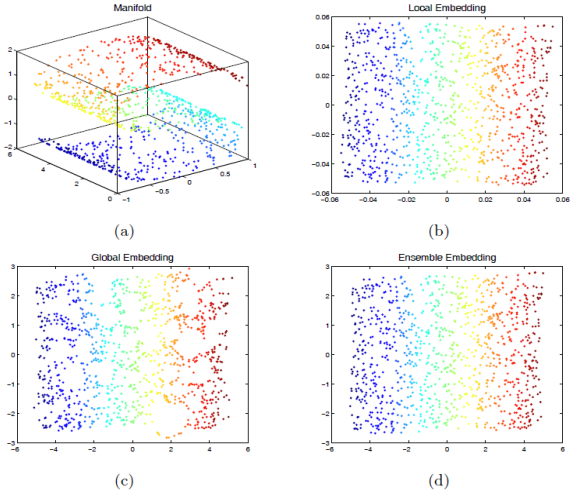


Fig. 2. Experimental results on the `SCurve` manifold. (a) The manifold. (b)-(d) Learning results by LTSA, ISOMAP, and MLEN, respectively.

we compare MLEN with its two components, LTSA and ISOMAP, and we set fixed parameters $\alpha = 0.1$ and $\varepsilon = 10^{-4}$.

Furthermore, we introduce two measures to quantitatively assess the learned embeddings. The global one is defined as the mean squared error between pairwise Euclidean distances in the embedding and ground truth geodesic distances on the

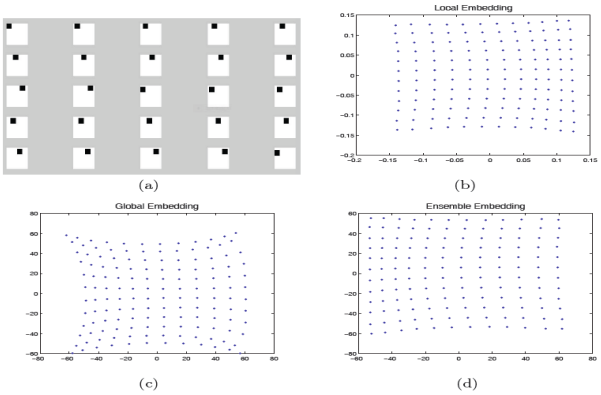


Fig. 3. Experimental results on the `image` manifold. (a) The manifold. (b)-(d) Learning results by LTSA, ISOMAP, and MLEN, respectively.

manifold. The local one is the LCMC criterion proposed in [4]. Both measures scale from 0 to 1, and a low value close to 0 indicates an embedding of good quality.

We first test MLEN on the SwissRoll and SCurve manifolds. For both manifolds, we randomly generate 1000 points as training data and set $k_l = k_g = 10$ and $d = 2$. The manifold and embeddings learned by LTSA, ISOMAP, and MLEN are shown in Figs. 1 and 2, respectively. Compared with ISOMAP, we can see that MLEN outputs a more regular embedding by visual inspection; while compared with LTSA, MLEN holds global aspect ratio. Quantitative assessments presented in Tab. 1 also validate the above observation.

We next test MLEN on a high-dimensional image manifold, which is generated by moving a black square of 16 by 16 pixels on white background with uniform steps. We obtain total 144 images and each image is of 64×64 pixels. The intrinsic degrees of freedom form a square grid in $\boldsymbol{R}^2$. Parts of the training images are shown in Fig. 3(a). We set $k_l = k_g = 12$ and $d = 2$. The learned embeddings are illustrated in Fig. 3, and quantitative assessments are summarized in Tab. 1. We can see that MLEN still outperforms its counterparts. It should be noted that since we do not have an explicit parameterization of this manifold, the ground truth geodesic distances are replaced with the approximated versions by ISOMAP.

4 Conclusion

In this paper, we proposed an ensemble method of manifold learners (MLEN), which takes local and global manifold learning methods, namely LTSA and ISOMAP, as its components. The embeddings learned by these sub-learners are fused by an ensemble learning process to output an overall embedding, which preserves both local

Table 1. Quantitative assessments on learned embeddings. Learning results with lowest scores, i.e., best quality, are marked in bold type.

Manifold	Measure	MLEN	ISOMAP	LTSA
SwissRoll	Global	**5.2831**	6.7245	1343.9
	Local	**0.0164**	0.1627	0.381
SCurve	Global	**0.0729**	0.0924	19.0084
	Local	**0.0186**	0.1165	0.1775
Image	Global	**77.1556**	85.9383	4266.2
	Local	**0.0378**	0.0617	0.0378

tangential coordinates and global geodesic distances. By introducing quantitative measures, we show that MLEN outperforms any of its components through experiments conducted on synthetic surfaces and image manifold.

Acknowledgment. This work was supported by the National Natural Science Foundation (NNSF) of China under Grant nos. 41201552 and no. 41174013.

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2007)
2. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting and variants. Machine Learning 36, 105–142 (1999)
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15(6), 1373–1396 (2003)
4. Chen, L., Buja, A.: Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. Journal of the American Statitical Association 104(485), 209–219 (2009)
5. Domingos, P.: A few useful things to know about machine learning. Communications of the ACM 55(10), 78–87 (2012)
6. Lin, T., Zha, H.: Riemannian manifold learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(5), 796–809 (2008)
7. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)
8. Tenenbaum, J.B., Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
9. Weinberger, K., Saul, L.: Unsupervised learning of image manifolds by semidefinite programming. International Journal of Computer Vision 70(1), 77–90 (2006)
10. Zhang, P., Ren, Y., Zhang, B.: A new embedding quality assessment method for manifold learning. Neurocomputing 97, 251–266 (2012)
11. Zhang, Z., Zha, H.: Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM Journal on Scientific Computing 26(1), 313–338 (2005)

Two Improved Artificial Bee Colony Algorithms Inspired by Grenade Explosion Method

Chaoqun Zhang[1,2,*], Jianguo Zheng[1], and Yongquan Zhou[2]

[1] Glorious Sun School of Business and Management, Donghua University, Shanghai, China
[2] College of Information Science and Engineering,
Guangxi University for Nationalities, Nanning, China
chaozi_0771@163.com

Abstract. In order to enhance the original artificial bee colony (ABC) algorithm's exploitation ability, two improved versions of ABC inspired by grenade explosion method (GEM), namely GABC1 and GABC2, are proposed. GEM is embedded in the employed bees' phase of GABC1, whereas it is embedded in the onlookers' phase of GABC2. The performance differences between GABC1 and GABC2 are assessed on five well-known benchmark functions and compared with that of ABC by analyzing the effect of different limit values. All the experimental results show that GABC2 greatly outperforms ABC on all the five functions. Although GABC1 has similar or better performance than GABC2 in most cases, GABC2 performs more robust and effective than GABC1.

Keywords: artificial bee colony algorithm, grenade explosion method, optimal search direction, exploitation ability.

1 Introduction

Artificial bee colony (ABC) algorithm first proposed by Karaboga [1] in 2005 simulates the foraging behavior of a honey bee swarm. Previous research [1-6] has shown that ABC is simple and more effective than other population-based algorithms. However, ABC is good at exploration but poor at exploitation [2].

Grenade explosion method (GEM) presented by Ahrari [7] in 2009 is an optimization technique, which inspired by the mechanism of a grenade explosion. Since GEM has high reliability and fast convergence [7-8], the method has attracted the attention of researchers all over the world in recent years [9].

Considering the disadvantages of ABC and advantages of GEM, two improved versions of ABC inspired by GEM, namely GABC1 and GABC2, are proposed. To the authors' knowledge, the present work is the first attempt to enhance the original ABC's exploitation ability based on GEM. GEM is embedded in the employed bees' phase of GABC1, whereas it is embedded in the onlookers' phase of GABC2. The performance differences between GABC1 and GABC2 are assessed on five

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 100–105, 2013.

well-known benchmark functions and compared with that of the original ABC by analyzing the effect of different limit values.

The rest of this paper is organized as follows. Section 2 introduces ABC. Section 3 describes the two proposed algorithms in detail. Section 4 compares and discusses the performance of ABC, GABC1 and GABC2. Finally, Section 5 presents conclusions.

2 Artificial Bee Colony Algorithm

In ABC, the colony of artificial bees contains three groups: employed bees, onlookers and scouts. The first half of the colony consists of employed bees and the second half includes onlookers. Employed bees search food sources and share the information about these food sources to recruit onlookers. Onlookers select the food sources found by all employed bees according to the probability proportional to the quality of food sources, and further exploit them. Scouts are translated from a few employed bees, which abandon their food sources through a predetermined number of cycles called *limit* and search new ones. The position of a food source represents a possible solution to an optimization problem, and the profitability of a food source corresponds to the fitness of the associated solution. Each food source is exploited by only one employed bee, namely, the number of food sources is equal to the number of employed bees.

At the beginning of an optimization, an initial population containing *SN* solutions is generated randomly. *SN* is the number of food sources, which is half of the population size(*NP*). Let $X_i=\{x_{i1},x_{i2},\ldots,x_{iD}\}$ represents the *i*th food source in the population, where *D* denotes the number of optimization parameters. And then, the population is subject to repeated cycles, $C=1,2,\ldots,$maximum cycle number(*MCN*), of the search processes of the employed bees, onlookers and scouts.

In ABC, the fitness function is defined as

$$fit(X_i)=\begin{cases}\dfrac{1}{1+f(X_i)}, f(X_i)\geq 0\\ 1+abs(f(X_i)), f(X_i)<0\end{cases} \tag{1}$$

where $f(X_i)$ is the objective function value of X_i, $fit(X_i)$ is the fitness value of X_i.

The probability of a food source being selected by an onlooker can be presented by

$$p(X_i)=\frac{fit(X_i)}{\sum_{n=1}^{SN} fit(X_n)} \tag{2}$$

After an employed bee discovers or an onlooker selects the food source X_i, they exploit a neighboring food source V_i. V_i is determined by changing only one parameter of X_i, namely $v_{ij}\neq x_{ij}$, while the rest of V_i keep the same value as X_i. v_{ij} is generated by

$$v_{ij}=x_{ij}+\phi_{ij}(x_{ij}-x_{kj}) \tag{3}$$

where $k\in\{1,2,\ldots,SN\}$ is a randomly chosen index and k must be different from i, $j\in\{1,2,\ldots,D\}$ is a randomly chosen dimension, ϕ_{ij} is a random number in the range

[-1,1]. Note that after an employed bee or onlooker determines a new candidate food source in the neighborhood of its currently associated food source using (3), a greedy selection mechanism [2-4] is applied between the new food source and the old one. If the abandoned food source is X_i, a scout produces a new food source according to

$$v_{ij} = x_{\min j} + rand(0,1)(x_{\max j} - x_{\min j}) \tag{4}$$

where x_{minj} and x_{maxj} are the lower and upper bounds of the variable x_{ij}, respectively. The main steps of ABC are outlined below:

Step1: Preset the values of parameters: *D*, *SN*, *MCN*, *limit*, *cycle*=1

Step2: Initialize and evaluate the population of food sources using (4) and (1)

Step3: repeat

Step4: Produce new food sources for the employed bees and evaluate them using (3) and (1), then apply the greedy selection process {employed bees' phase}

Step5: Calculate the probability values for food sources using (2)

Step6: Produce new food sources for the onlookers from the food source X_i selected depending on $p(X_i)$ and evaluate them using (3) and (1), then apply the greedy selection process {onlookers' phase}

Step7: Determine the abandoned food source for the scout, if exists, and replace it with a new randomly produced one using (4) {scouts' phase}

Step8: Memorize the best food source achieved so far

Step9: *cycle*=*cycle*+1

Step10: until *cycle*=*MCN*

3 Two Improved ABC Algorithms Inspired by GEM

In ABC, after an employed bee or onlooker selects the food source X_i, they exploit a neighboring food source V_i which is determined by changing only one parameter of X_i, namely $v_{ij} \neq x_{ij}$. v_{ij} is generated by (3). In (3), v_{ij} is modified from x_{ij} based on a comparison with the randomly selected position from x_{kj}. Obviously, j is a crucial parameter since it directly influences the position of a new food source. However, the randomly chosen dimension j may not always guide ABC toward more high fitted positions of food sources and lead to slow convergence. Then which dimension among all the dimensions is the best choice for an employed bee or onlooker to update the new candidate food source? GEM [7-8] is inspired by the mechanism of a grenade explosion, where objects are hit by pieces of shrapnel. Damage caused by each piece of shrapnel hitting an object is calculated. A high value for damage-per-piece in an area indicates there are valuable objects in that area. To intensify the damage, the next grenade is thrown where the greatest damage occurs. This process would result in finding the best place for throwing the grenade. Therefore, we introduce GEM into ABC called GABC1 and GABC2, respectively, to select the optimal search dimension instead of a random chosen dimension for each employed bee or onlooker. Here, the overall damage caused by the hit is considered as the fitness of a solution. For the sake of simplification, we just choose one grenade and *D* pieces of shrapnel in each iteration of GABC1 and GABC2.

In each iteration of GABC1 or GABC2, D pieces of shrapnel are thrown in all the dimensions to gather information around the current position of the grenade, meanwhile, each employed bee or onlooker computes each candidate food source along which each shrapnel is thrown and evaluates its damage-per-shrapnel value, then makes a decision on a new candidate food source with the greatest damage. Consequently, in GABC1 or GABC2, a new candidate solution based on the optimal search dimension for an employed bee or onlooker is produced by:

$$v_{iOSD} = x_{iOSD} + \phi_{iOSD}(x_{iOSD} - x_{kOSD}) \tag{5}$$

$$s.t. \quad fit(V_{iOSD}) = \max\{ fit(V_{it}) \mid t = 1,2,\cdots,D\} \tag{6}$$

where $k \in \{1,2,\ldots,SN\}$ is a randomly chosen index and $k \neq i$; $OSD \in \{1,2,\ldots,D\}$ and OSD is the optimal search dimension determined by (6); ϕ_{iOSD} is a random number in the range [-1,1]; V_{it} denotes the new candidate food source V_i generated by just changing the value of the old food source X_i in dimension t, namely $v_{it} \neq x_{it}$, while the rest of V_{it} keep the same value as X_i; V_{iOSD} has a similar meaning as V_{it} and also indicates V_i obtains the maximum fitness in dimension OSD instead of other dimensions.

From the above explanation, the essential difference between GABC1 and ABC is different exploitation strategy adopted by their employed bees, whereas the essential difference between GABC2 and ABC is different exploitation strategy adopted by their onlookers. Therefore, the main steps of GABC1 remain the same as ABC except for Step 4 listed below (the main differences from ABC are highlighted in bold):

Step4: Produce and evaluate new food sources for each employed bee in all dimensions of each associated food source and determine the optimal search dimension(OSD) and the best new candidate food source using (5), (6) and (1) , then apply the greedy selection process {employed bees' phase}

Meanwhile, the main steps of GABC2 remain the same as ABC except for Step 6 presented below (the main differences from ABC are highlighted in bold):

Step6: Produce and evaluate new food sources for each onlooker in all dimensions of each associated food source according to its probability $p(X_i)$ and determine the optimal search dimension(OSD) and the best new candidate food source using (5), (6) and (1) , then apply the greedy selection process {onlookers' phase}

4 Experiments and Discussion

The performance of population-based meta-heuristic greatly depends on the control parameters [10]. ABC has only one control parameter (namely *limit*) apart from the common control parameters of the population-based algorithms [2]. Furthermore, it is a common practice to compare different algorithms using different benchmark problems in the field of optimization [9].Therefore, the performance differences between GABC1 and GABC2 are assessed on the five well-known functions given in [3] and compared with that of ABC by analyzing the effect of different limit values. In the literature [2,4], NP,D and *limit* were often set to 20,100 and $SN \times D$,

respectively. Thereby, in all experiments, we fixed *SN*,*D*,*MCN* as 10,100,7000, and set *limit* to 0.01×S*N*×*D*,S*N*×*D*,100×S*N*×*D*, namely, 10,1000,100000, respectively. GABC1 and GABC2 were coded in MATLAB version 7.0.4.365. The MATLAB codes of ABC were downloaded from http://mf.erciyes.edu.tr/abc/software.htm. All experiments were run on a portable computer with an Intel 1.70GHz CPU and 4GB RAM under Windows 7. In the three algorithms, the number of scouts was selected as one. All experiments were repeated 30 times. The execution time of 30 runs and mean of the function values found by the three algorithms under the same conditions are given in Table 1.

Table 1. Results obtained by ABC,GABC1 and GABC2 with different limit values.

Function	*limit*	ABC		GABC1		GABC2	
		Mean	Time(s)	Mean	Time(s)	Mean	Time(s)
Griewank	10	58.1256	260	3.57E-15	1661	2.05E-04	1647
	1000	0.0011	247	2.74E-15	1617	2.78E-15	1546
	100000	0.0012	245	3.26E-15	1524	3.18E-15	1498
Rastrigin	10	939.4184	198	8.98E-13	1007	0.0127	1060
	1000	0.0416	185	6.14E-13	961	6.86E-13	869
	100000	0.0852	185	0.2653	887	7.39E-13	858
Rosenbrock	10	1.31E+05	193	51.1141	572	30.1412	548
	1000	0.7834	192	18.4045	570	0.2197	545
	100000	0.9636	192	19.1806	571	0.2378	546
Ackley	10	15.6603	218	1.78E-13	1029	1.59E-13	999
	1000	1.62E-12	202	1.43E-13	974	1.35E-13	953
	100000	1.91E-12	203	1.56E-13	875	1.47E-13	847
Schwefel	10	1.83E+04	202	0.0013	1755	0.0017	1738
	1000	442.7520	199	0.0013	1751	0.0013	1723
	100000	464.8320	199	7.8972	1745	3.9492	1721

As seen from Table 1, under the five limit values, each execution time of the three algorithms on the same function is almost the same, and both the proposed algorithms need a longer execution time than ABC. However, GABC1 and GABC2 obviously performed more stable and reliable than ABC. We can observe that the three algorithms have the best results under *limit*=1000 in all the cases. Besides, very low or large limit values such as 10,100000 might worsen the performance of the three algorithms. For example, the results obtained by GABC2 with the five limit values are the same order on each function except for Griewank,Rastrigin,Rosenbrock functions under *limit*=10 and Schwefel under *limit*=100000, while ABC performed poorly, especially on Schwefel under the five limit values and the rest four functions under *limit*=10. Besides, GABC1 has similar or better performance than GABC2 on Griewank,Rastrigin and Schwefel except for them under *limit*=100000. However, because ABC and GABC2 can select more random solutions in the employed bees' phase than GABC1, GABC1 sometimes performs worse than GABC2 and ABC, especially on Rosenbrock. These show that GABC2 performs more robust and effective than GABC1.

The above results are reasonable. On the one hand, GABC1, GABC2 or ABC modifies only one parameter of a solution in each cycle and at least *D* cycle is needed to change all parameters of one solution. Hence a value around *SN*×*D* is generally

appropriate for *limit*. On the other hand, there is an inverse proportionality between the value of *limit* and the scout production frequency. In other words, lower limit values cause more scouts to be produced than needed while higher limit values cause scouts not to occur often and this attenuates the exploration ability of the three algorithms.

5 Conclusions

Two improved versions of ABC inspired by GEM, namely GABC1 and GABC2, are proposed to enhance the original ABC's exploitation ability. All the experimental results show that GABC2 greatly outperforms ABC on all the five functions. Although GABC1 has similar or better performance than GABC2 in most cases, GABC2 performs more robust and effective than GABC1. These demonstrate two facts: On the one hand, the exploitation strategy adopted by employed bees or onlookers has positive effect on the performance of GABC1 and GABC2; On the other hand, it is necessary for an algorithm to provide adequate random solutions in early iterations so as to maintain the population diversity.

Acknowledgement. This work was supported by National Science Foundation of China (Grant Nos.70971020 and 61165015), Open Research Fund of Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis (Grant No. 2012HCI09) and Key Project of Guangxi University for Nationalities (Grant No. 2012MDZD035).

References

1. Karaboga, D.: An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical report, Erciyes University (2005)
2. Akay, B., Karaboga, D.: A Modified Artificial Bee Colony Algorithm for Real-parameter Optimization. Inform. Sciences 192, 120–142 (2012)
3. Karaboga, D., Basturk, B.: A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. J. Global Optim. 39, 459–471 (2007)
4. Karaboga, D., Basturk, B.: On the Performance of Artificial Bee Colony(ABC) Algorithm. Appl. Soft Comput. 8, 687–697 (2008)
5. Karaboga, N., Latifoglu, F.: Adaptive Filtering Noisy Transcranial Doppler Signal by Using Artificial Bee Colony Algorithm. Eng. Appl. Artif. Intel. 26, 677–684 (2013)
6. Chang, W.D.: Nonlinear CSTR Control System Design Using an Artificial Bee Colony Algorithm. Simul. Model. Pract. Th. 31, 1–9 (2013)
7. Ahrari, A., Shariat-Panahi, M., Atai, A.A.: GEM: A Novel Evolutionary Optimization Method with Improved Neighborhood Search. Appl. Math. Comput. 210, 376–386 (2009)
8. Ahrari, A., Atai, A.A.: Grenade Explosion Method–a Novel Tool for Optimization of Multimodal Functions. Appl. Soft Comput. 10, 1132–1140 (2010)
9. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching-learning-based Optimization: An Optimization Method for Continuous Non-linear Large Scale Problems. Inform. Sciences 183, 1–15 (2012)
10. Wu, B., Qian, C., Ni, W., Fan, S.: The Improvement of Glowworm Swarm Optimization of Continuous Optimization Problems. Expert Syst. Appl. 39, 6335–6342 (2012)

3D Protein Structure Prediction with Local Adjust Tabu Search Algorithm

Xiaoli Lin and Fengli Zhou

Information and Engineering Department of City College,
Wuhan University of Science and Technology Wuhan, China
aneya@163.com, Fenglizhou@yahoo.cn

Abstract. The protein folding structure prediction is computationally challenging and has been shown to be *NP*-hard when the 3D off-lattice *AB* model is employed. In this paper, the local adjustment tabu search (LATS) algorithm has been used to search the ground state of 3D *AB* off-lattice model for protein folding structure. A kind of optimization about the neighborhood scale and the annealing mechanism has been presented, where a local adjustment strategy has also been used to enhance the searching ability for the global minimum within the *AB* off-lattice model. Experimental results demonstrate that the proposed algorithm has better performance in global optimization and can predict 3D protein structure more effectively.

Keywords: Protein Structure Prediction, Tabu Search Algorithm, 3D Off-Lattice Model, Local Adjustment.

1 Introduction

Protein structure prediction is one of the most challenging objectives in bioinformatics [1]. Because of the complexity of the realistic protein structure, it is hard to determine the exact tri-dimensional structure from its sequence of amino acids [2]. Therefore, a lot of simplified protein structure models have been developed. Hydrophobic-polar (*HP*) model [3] has become one of the major tools for studying protein structure [4]. However, the *HP* lattice-model doesn't reveal all secrets of the protein [5].

Currently, *AB* off-lattice model has been widely applied to protein structure prediction. Stillinger [6] studied the off-lattice *AB* protein model in two dimensions which uses only two types of residues. Bachmann, etc. [7] studied the off-lattice *AB* model using the energy landscape paving minimizer (ELP). It is found that their results for the ground state energies are lower than the best values reported in the earlier literature. Kim, etc. [8] also studied the same model by the conformational space annealing (CSA), and obtained even better results. Recently, researchers have developed many algorithms to predict protein, such as pruned-enriched-Rosenbluth method (PERM) [9], particle swarm optimization (EPSO) [10] etc.

In this paper we study an extension to three dimensions (3D) of a two dimensions (2D) off-lattice model [6] that was successfully used in [11-12]. The 3D model we used has been introduced in [13-14], which considered the torsional energy implicitly.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 106–111, 2013.

In order to accurately search for the ground-state conformations of the protein, an improved hybrid algorithm that combines tabu search algorithm and simulated annealing was applied.

2 *AB* Off-Lattice Model

The off-lattice *AB* model consists of hydrophobic *A* monomers ($\sigma_i = +1$) and hydrophilic *B* monomers ($\sigma_i = -1$). The energy function is given by [5]

$$E = -k_1 \sum_{i=1}^{N-2} \hat{b}_i \cdot \hat{b}_{i+1} - k_2 \sum_{i=1}^{N-3} \hat{b}_i \cdot \hat{b}_{i+2} + \sum_{i=1}^{N-2} \sum_{j=i+2}^{N} E_{LJ}(r_{ij}; \sigma_i, \sigma_j) \tag{1}$$

Where $\hat{b}_i$ is the bond vector between the monomers i and $i+1$ with unit length. Since $\hat{b}_i \cdot \hat{b}_{i+1} = \cos\theta_i$, $\hat{b}_i \cdot \hat{b}_{i+2} = \cos\alpha_i$, the N -mer can be specified by the $N-1$ bond vectors $\hat{b}_i$ or by $N-2$ bond angles θ_i and $N-3$ torsional angles. The r_{ij} depends on the bond angles and torsional angles. These two angles are the degrees of freedom of the model. The species-dependent global interactions are given by

$$E_{LJ}(r_{ij}; \sigma_i, \sigma_j) = 4C(\sigma_i, \sigma_j)\left(\frac{1}{r_{ij}^{12}} - \frac{1}{r_{ij}^{6}}\right) \tag{2}$$

3 Methods Description

Tabu search proposed and formalized by Glover [15] is an intelligent heuristic search procedure. Tabu search is designed to explore the solution space beyond local optimality. It uses an operation that changes the current solution and allows to visiting a neighborhood of the given current solution. One of the main components of tabu search is the use of adaptive memory, that is, local choices are guided by the past history of the search. Adaptive memory helps the search process to avoid local optima and explores the solution space economically and effectively [16].

3.1 Adaptive Neighborhood Generation

The generation of neighborhood solution is an important step for getting a final improved result. A good neighborhood solution effectively converges faster so it cuts the computational cost, while the scale of neighborhood is a critical to generate neighborhood solution. In this paper, an optimization strategy about neighborhood scale is proposed.

In the initial phase, the scale of neighborhood must be sufficiently large to avoid falling into the premature convergence and becoming trapped in a local optimum.

While in the later phase of the process, the scale of neighborhood must be sufficiently small to avoid missing the best solution. So define a linearity combination:

$$\delta X = TCurrent * Definition / TStart \tag{3}$$

Where δX is neighborhood, $TCurrent$ is current temperature, $TStart$ is initial temperature. The neighborhood will become smaller gradually with the drop of temperature.

3.2 Annealing Mechanism

Simulated Annealing (SA) [17] is very powerful in solving complicated problems. As cooling proceeds, the system becomes more ordered and approaches a frozen ground state at $T = 0$. Hence the process can be considered as an adiabatic approach to the lowest energy state [18]. In this paper, the cooling schedule that provides necessary and sufficient conditions for convergence is

$$T_{i+1} = \sigma T_i (0 \le \sigma \le 1) \tag{4}$$

This is a simple linear equation. When σ inclines to 1, the cooling speed of temperature becomes increasingly slow.

3.3 Local Adjustment Strategy

The local adjustment strategy is proposed to enhance the searching ability when searching for optimum solutions within the *AB* off-lattice model.

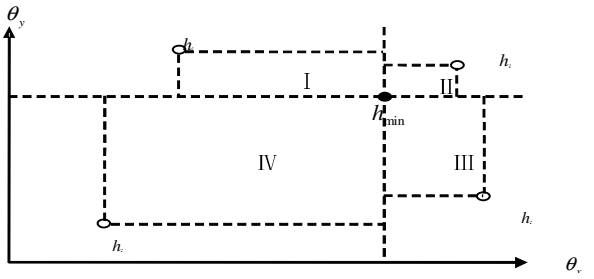


Fig. 1. A schematic diagram of local adjustment

Fig. 1 presents the schematic diagram of local adjustment for the tetramer. According to *AB* off-lattice model, the energy $\Phi(\theta)$ of each tetramer is determined by two angles of bend θ_x and θ_y. Assume that two parameters $\theta_x^{\min}$ and $\theta_y^{\min}$ of $h_{\min}$ separate θ_x-θ_y coordinate domain into 2^2 areas, the distance between each parameter of each individual h_i and the corresponding parameter of $h_{\min}$ can be denoted as

$$\begin{cases} \Delta\theta_x = \theta_x^i - \theta_x^{\min} \\ \Delta\theta_y = \theta_y^i - \theta_y^{\min} \end{cases} \tag{5}$$

Since searching smaller areas in which the global minimum lies will lead to a faster convergence to the desired solution, every parameter of h_i is adjusted by (6) to keep the searching in smaller areas.

$$\begin{cases} \theta_x^i = \theta_x^i + Rand\ \ (0..1)\Delta\theta_x \\ \theta_y^i = \theta_y^i + Rand\ \ (0..1)\Delta\theta_y \end{cases} \tag{6}$$

3.4 The Algorithm

The general idea of LATS based on the *AB* off-lattice model is followed. The search algorithm starts with the initialization of parameters by some appropriate values. Then, the population is generated randomly, and the energy of each individual in neighborhood list is calculated according to the potential-energy function (1). Then neighborhood solutions are rearranged from minimal energy to maximal energy, and some individuals are selected as candidate solutions. In the process of annealing, the deprecated principle and the local adjustment strategy are employed continuously. The algorithm is described as follows.

```
LATS(ns, cs, st, et, dt)
{ Initialize: BestSoFar • Create hypotheses x at random
  T = TStart
  While (T*dT > TStop) do
    For( loopCounter = 0, loopCounter++ < innerLoopTimes)
    { Generate neighborhood solutions;
      calculate the energy value Model.CountValue(h);
      Select candidate solutions from neighborhood List;
      Use deprecated principle;
      Add the best solution to the Tabu List;
      Apply the local adjustment strategy; }
}
```

4 Optimization Results

This part uses the same Fibonacci sequences in [6] for comparison. Table 1 presents the experimental results of the previous algorithms, as well as those of LATS. E_{ACMC} is the minimum energy obtained by the annealing contour Monte Carlo (ACMC) algorithm [19] while E_{ELP} and E_{CSA} are the minimum energy obtained by the energy landscape paving minimizer ELP [7] and the conformational space annealing (CSA) algorithm [8] respectively. E_{LAGA} is the minimum energy obtained by the local adjust genetic algorithm introduced in [13].

It can be seen that the results obtained with LATS are smaller than those of the CSA for all the four sequences. For length 13, our result is equal to that of ACMC, and is smaller than other result. For length 21, our result is slightly smaller than that of ACMC and CSA, and equal to those of ELP and LAGA. For other cases, our results are smaller than the results obtained by other methods, which shows that LATS has better performance for long sequence.

Fig. 2 depicts the lowest energy conformations in the 3D off-lattice *AB* model obtained by LATS algorithm, where red balls represent hydrophobic *A* monomers, grey balls represent hydrophilic *B* monomers. In Fig. 2, the conformations form a single hydrophobic core for four Fibonacci sequence. According to the phenomenon that hydrophobic amino acids fold into a hydrophobic core surrounded by hydrophilic amino acids in protein molecule, it indicates that the 3D off-lattice *AB* model appears to be an effective way to simulate the protein folding. Comparing with 2D off-lattice *AB* model, this model is more like real protein structure.

Table 1. The minimum energies obtained for Fibonacci sequences with $13 \le N \le 55$

LENGTH	SEQUENCE	E_{ELP}	E_{ACMC}	E_{CSA}	E_{LAGA}	E_{LATS}
13	ABBABBABABBAB	-26.498	-26.507	-26.471	-26.498	-26.507
21	BABABBABABBABBABABBAB	-52.917	-51.757	-52.787	-52.917	-52.917
34	ABBABBABABBABBABABBAB ABBABBABABBAB	-92.746	-94.043	-97.732	-98.765	-99.876
55	BABABBABABBABBABABBAB ABBABBABABBABBABABBAB ABBABBABABBAB	-172.696	-154.505	-173.980	-176.542	-178.986

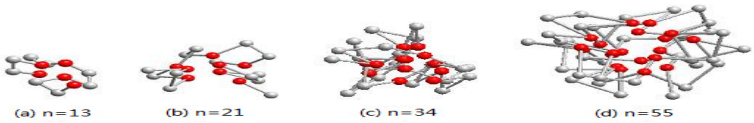


Fig. 2. The lowest energy conformations for the four Fibonacci sequences obtained by LATS

5 Conclusions

The 3D *AB* off-lattice model is a great improvement of simplification model of protein folding. This paper describes an improved local adjustment tabu search algorithm for protein folding structure prediction. The novel strategies are employed to improve the searching lowest energy conformation of the protein. The experimental results show LATS has the advantage over the previous algorithms. But this 3D off-lattice *AB* model only considers two kinds of residues and two kinds of interaction energy, so we should improve the algorithm and make it more effective for more properties of amino acid residues and more interaction energy.

References

1. Anfinsen, C.B.: Principles that govern the folding of protein chains. Science 181, 223–227 (1973)
2. Lopes, H.S.: Evolutionary Algorithms for the Protein Folding Problem: A Review and Current Trends. In: Smolinski, T.G., Milanova, M.G., Hassanien, A.-E. (eds.) Computational Intelligence in Biomedicine and Bioinformatics. SCI, vol. 151, pp. 297–315. Springer, Heidelberg (2008)
3. Dill, K.A.: Theory for the folding and stability of globular proteins. Biochemistry 24, 1501–1509 (1985)
4. Hart, W.E., Newman, A.: Protein structure prediction with lattice models. In: Aluru, S. (ed.) Handbook of Molecular Biology. Computer and Information Science Series, pp. 1–24. Chapman & Hall/CRC Press (2006)
5. Irbäck, A., Peterson, C., Potthast, F., Sommelius, O.: Local interactions and protein folding: A three-dimensional off-lattice approach. J. Chem. Phys. 107, 273–282 (1997)
6. Stillinger, F.H.: Collective aspects of protein folding illustrated by a toy model. Phys. Rev. E 52, 2872–2877 (1995)
7. Bachmann, M., Arkin, H., Janke, W.: Multicanonical study of coarse-grained off-lattice models for folding heteropolymers. Phys. Rev. E 71, 31906 (2005)
8. Kim, S.-Y., Lee, S.B., Lee, J.: Structure optimization by conformational space annealing in an off-lattice protein model. Phys. Rev. E 72, 011916 (2005)
9. Hsu, H.-P., Mehra, V., Grassberger, P.: Structure optimization in an off-lattice protein model. Phys. Rev. E 68, 037703 (2003)
10. Zhu, H.B., Pu, C.D., Lin, X.L.: Protein structure prediction with EPSO in toy model. In: 2009 Second International Conference on Intelligent Networks and Intelligent Systems (2009)
11. Lin, X.L., Zhu, H.B.: Structure Optimization by an Improved Tabu Search in the AB Off-Lattice Protein Model. In: The 1st International Conference on Intelligent Networks and Intelligent Systems, pp. 123–126 (2008)
12. Lin, X.L., Yu, Z.H.: Effective Protein Folding Structure Prediction with the Local Adjustment Tabu Search Algorithm. In: ICMAI (2012)
13. Zhang, X.L., Lin, X.L.: Effective 3D Protein Structure Prediction With Local Adjustment Genetic-Annealing Algorithm. Interdiscip. Sci. Comput. Life Sci. 2, 1–7 (2010)
14. Zhang, X.L., Lin, X.L., Wan, C.P., Li, T.T.: Genetic-annealing algorithm for 3D off-lattice protein folding model. The 2nd BioDM Workshop on Data Mining for Biomedical Applications, PAKDD Workshops, 186–193 (2007)
15. Turkensteen, M., Andersen, K.A.: A Tabu Search Approach to Clustering. In: Operations Research Proceedings (2008)
16. Tariq, R., Yi, W., Kuo-Bin, L.: Multiple Sequence Alignment Using Tabu Search. In: Proc. Second Asia-Pacific Bioinformatics Conference (2004)
17. Lecchini-Visintini, A., Lygeros, J., Maciejowski, J.: Simulated Annealing: Rigorous finite-time guarantees for optimization on continuous domains. In: Advances in Neural Information Processing Systems 20, Proceedings of NIPS (2007)
18. Gatti, C.J., Hughes, R.E.: Optimization of Muscle Wrapping Objects Using Simulated Annealing. Annals of Biomedical Engineering 37, 1342–1347 (2009)
19. Liang, F.: Annealing contour Monte Carlo algorithm for structure optimization in an off-lattice protein model. J. Chem. Phys. 120, 6756 (2004)

An Effective Parameter Estimation Approach for the Inference of Gene Networks

Yu-Ting Hsiao and Wei-Po Lee

Department of Information Management
National Sun Yat-sen University
Kaohsiung, Taiwan
wplee@mail.nsysu.edu.tw

Abstract. Constructing genetic regulatory networks from expression profiles is one of the most important issues in systems biology research. To automate the procedure of network construction, this work presents an integrated approach for network inference, in which the parameter identification and parameter optimization techniques are developed to deal with the scalability and network robustness problems, respectively. To validate the proposed approach, experiments have been conducted on several artificial and real datasets. The results show that our approach can be used to infer robust gene networks with desired system behaviors successfully**.**

Keywords: gene network inference, parameter identification, parameter optimization, structural information.

1 Introduction

Constructing genetic regulatory networks (GRNs) from expression data is one of the most important issues in systems biology research, and reverse engineering has been advocated to reconstruct networks in an automated way [1]. In the process of inferring gene network from gene expression data, many computational models have been proposed to address different levels of biological details, ranging from the very abstract to the very concrete, depending on the biological level to be studied. In this work we adopt one of the most popular and well-researched concrete models, the S-system model, to represent a gene network. S-system is a type of ordinary differential equation model, in which the component processes are characterized by power law functions [2][3]. In the S-system model, the systematic structure can be described as:

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{N} x_j^{g_{i,j}} - \beta_i \prod_{j=1}^{N} x_j^{h_{i,j}}$$

Where x_i is the expression level of gene i and N is the number of genes in a genetic network. The non-negative parameters α_i and β_i are rate constants that indicate the direction of mass flow. The real number exponents $g_{i,j}$ and $h_{i,j}$ are kinetic orders that reflect the strength of interactions from gene j to i. To infer an S-system model is, therefore, to determine all of the $2N(N+1)$ parameters simultaneously.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 112–117, 2013.

One major goal in gene network reconstruction is to minimize the accumulated discrepancy between the gene expression data recorded in the data set and the values produced by the inferred model. The performance of a certain model can thus be defined directly as the mean squared error over the time period:

$$\sum_{i=1}^{N}\sum_{t=1}^{T}\left\{\frac{x_i^a(t)-x_i^d(t)}{x_i^d(t)}\right\}^2$$

In the above equation, $x_i^d(t)$ is the desired expression level of gene i at time t, $x_i^a(t)$ is the value generated from the inferred model, N is the number of genes in the network, and T is the number of time points in measuring gene expression data.

In the procedure of inferring networks, the first challenge is to deal with the *scalability* problem: it is difficult to determine the large number of parameters involved in a GRN. Another challenge is to determine the final solution with a correct network structure, as the network inference is in fact an under-determined problem. To solve the structure problem, some researchers suggested that to directly take the form of parameter constraint for priori domain knowledge to restrict the parameter search (e.g., [3]). Others proposed to limit the number of the GRN connectivity as small as possible, for example, [2].

Prior knowledge, however, is not always available. Though some tools can be used to derive skeletal network structures from time-series data, for example BoolNet [4], their results are not accurate to work as structural knowledge to guide the search. Here, we propose to turn to infer robust networks that are highly possible to have correct structures. In this work, we present an integrated approach to iteratively evolve partial solutions to guide the search toward the target complete solution gradually. This approach, to our knowledge, is the first work considering both internal and external network characteristics for network inference. To validate the proposed approach, a series of experiments have been conducted on artificial and real datasets. The results and analysis show that our approach can infer robust networks with desired system behavior successfully from the gene profiles.

2 An Integrated Approach for Gene Network Inference

The proposed approach includes two procedures for parameter identification and parameter optimization, respectively. As the recent surveys of population based algorithms have revealed that the hybrid methods of GA and PSO can lead to better results in solving optimization problems than the individual methods alone, therefore, here we extend the hybrid method we developed previously ([3][5]) to cope with the parameter sensitivity. The second procedure, parameter identification, is developed to work with the optimization procedure. This procedure is performed when the evolution proceeds to a predefined number of iterations. It mainly includes a sensitivity analysis method to calculate the parameter sensitivity, selects the most sensitive network parameters and determines value ranges for them (to work as implicit structural knowledge), and then sends the parameters with their value constraints back to the optimization procedure to continue the search.

The parameter identification procedure mentioned above is based on a sensitivity analysis (SA) technique. With the parameter sensitivity, we can identify the critical parameters or discover unimportant parameters that may have a positive or negative influence on a network, by varying the parameter values within a certain range and performing statistical calculations to measure the system instability. In the case of inferring a gene network from the expression data, the network structure is generally unclear during the modeling process. It is thus not possible to identify and select the most important parameters to perform the global sensitivity analysis that is widely used in the studies of dynamic systems. To consider multiple network parameters simultaneously, we devise a new approach modified from a widely used SA technique, multi-parameter sensitivity analysis (MPSA, [6]). The proposed method *m*-MPSA takes an iterative process to calculate the sensitivity of each parameter, and then ranks the sensitivities of all parameters. By considering the most sensitive parameters first, the algorithm can then infer robust solutions.

In addition to the above parameter identification procedure, the hybrid GA-PSO procedure for parameter optimization also plays an important role in the proposed approach. In our implementation, we take a direct encoding scheme to represent solutions for both GA and PSO parts, in which the network parameters (i.e., α_i, β_i, $g_{i,j}$, and $h_{i,j}$ in the S-system model) are arranged as a linear string chromosome of floating-point numbers. And the fitness function is defined as the error function described in section 1 for performance measurement.

The optimization procedure operates as the following. Initially, a population is randomly generated and evaluated. The individual solutions are ranked according to their fitness values, and then divided into two parts: (1-*r*)% and *r*% of the population for running PSO and GA, respectively. The best individuals (including (1-*r*)% of the whole population) are preserved and enhanced by the PSO procedure. They are then sent to the next generation. Meanwhile, the remaining individuals (including *r*% of the population) are discarded. To replace the removed individuals, the tournament selection scheme is used to choose the same number of individuals (i.e., *r*% of the population) from the best ones (before they are updated by PSO), and the selected individuals are used to create new individuals by GA. The details of the parameter optimization steps are referred to our original work [3].

3 Experiments and Results

In this section, we describe how we conducted a series of experiments to verify the developed integrated approach from two different perspectives: the external network behavior and the internal network robustness. We first examined the performance of applying computational methods with *m*-MPSA to artificial datasets, and then focused on the evaluation and analysis for two real-world datasets. Due to the space limitation, we only report two sets of results as representative examples.

In the first set of experiments, the example is a five-node network taken from [7], in which the nodes have the following non-linear relationships:

$$\dot{X}_1 = 15.0X_3X_5^{-0.1} - 10.0X_1^{2.0}$$
$$\dot{X}_2 = 10.0X_1^{2.0} - 10.0X_2^{2.0}$$
$$\dot{X}_3 = 10.0X_2^{-0.1} - 10.0X_2^{-0.1}X_3^{2.0}$$
$$\dot{X}_4 = 8.0X_1^{2.0}X_5^{-0.1} - 10.0X_4^{2.0}$$
$$\dot{X}_5 = 10.0X_4^{2.0} - 10.0X_5^{2.0}$$

To collect time series data, we started and continued network operations for thirty simulation steps. Before using the developed approach to infer robust results, we conducted an investigation on the above dataset to examine the effect of considering network structure in the inference procedure. Here, a small penalty on structure correctness was added to the fitness function given in section 1, and a weighting factor was used for summing the two types of penalty (i.e., the behavior error and the structure error). Table 1 shows the results (averaged from twenty runs) of using different weighting ratios. As can be seen, when the structure error was introduced to the evaluation function, the resulting model tended to have a structure closer to the original network, but a less-fitted behavior. This table shows the importance of network structure and indicates the need of an inference approach to take both issues into account.

Table 1. Effect of considering structure correctness in the evaluation function

behavior : structure	(GA-PSO)	2:8	3:7	5:5	7:3	8:2
fitness (avg)	0.0589	0.2848	0.2560	0.2523	0.1661	0.1072
structure (avg)	28.33%	77.78%	64.44%	64.11%	56.11%	58.89%

With the above results, we continued to perform experiments to evaluate our approach that considers network robustness as a factor representing the unknown network structure. Two optimization algorithms (i.e., PSO and GA-PSO) were arranged, and twenty independent runs were conducted for each arrangement. Table 2 shows the results. From Table 2, we can observe that both computational methods with SA consistently outperform the ones without using SA in networks inference, and the proposed GA-PSO method performed better than the traditional PSO method. To evaluate the robustness of the evolved gene networks, we also list the sensitivity values of the best solutions obtained from the final generations. They show that the runs with SA were able to evolve networks with lower fitness values and lower sensitivity values (i.e., more robust networks) simultaneously.

Table 2. Fitness and sensitivity obtained by different settings

	PSO		GA-PSO	
	w/o SA	with SA	w/o SA	with SA
Avg	0.5718	0.3647	0.0589	0.0192
Best	0.3172	0.1315	0.0310	0.0098
Worst	0.8492	0.4928	0.1034	0.0314
S.D	0.1729	0.1286	0.0229	0.0062
Sensitivity	0.7334	0.7002	0.8249	0.7581

After evaluating the performance of our SA-based approach in network inference, we conducted a set of experiments to investigate how our approach can be applied to the studies of real gene networks. As the GA-PSO algorithm has been shown to outperform the other method, here we present the results of using this method on the example of real-world data. This dataset comes from a study of the SOS DNA repair system in *E. coli.* Here, we chose six genes from the original experimental data reported in [8] for our study, because the interactions of these six genes have been well studied and commonly used in the related works (e.g., [9][10]). Though there have been several studies on inferring the SOS repair system, the most related works adopted a decoupled S-system model [11][12]. Therefore we also used decoupled differentials to describe gene profiles. In a decoupled S-system, a tightly coupled system of non-linear differential equations is decomposed into several differential equations, each of which describes a specific gene that can be separately inferred. For the de-coupled model described above, the evaluation function of the *i*-th sub-problem can thus be given as the following:

$$f = \sum_{t=1}^{T} \left\{ \frac{x_i^a(t) - x_i^d(t)}{x_i^d(t)} \right\}^2$$

Where $x_i^d(t)$ is the desired expression level of gene i at time t, $x_i^a(t)$ is the value generated from the inferred model, and T is the number of time points in measuring gene profiles.

With the above evaluation function, we employed our integrated approach of parameter identification and parameter optimization to infer gene interactions. Table 3 presents the sensitivity values obtained by different settings. Again, the results show that the proposed approach can obtain better results than the original algorithm for all six genes. It means that the network inferred by the proposed approach is more robust and thus has a correct network structure.

Table 3. Results for the SOS dataset

	lexA		uvrA		uvrD	
	w/o SA	with SA	w/o SA	with SA	w/o SA	with SA
sensitivity	0.8355	0.7977	0.8417	0.7838	0.7890	0.7890
	recA		umuD		polB	
	w/o SA	with SA	w/o SA	with SA	w/o SA	with SA
sensitivity	0.8669	0.7943	0.8315	0.7841	0.8507	0.7787

4 Conclusion

In this work, we have indicated the importance of considering both the network behavior and the network structure in inferring gene networks. We have also developed an integrated approach that takes network robustness into account in the inference procedure, when the structural information of the network is not available. Our approach includes two parts. The first part is a parameter identification procedure that involves a sensitivity analysis method to select sensitive parameters and to derive

upper and lower bounds for these parameters. And the second part, a parameter optimization procedure that takes the bounds to infer gene networks. In this way, the inferred networks can be robust and have desired behaviors at the same time. To validate the proposed approach, a series of experiments have been conducted. The results show that our integrated approach outperformed others. Especially, in our experiments, we have further analyzed extensively the results obtained from the real gene datasets of SOS repair system. The results confirm that the proposed approach can successfully explore the critical parameters for genes involved in the biological systems in practice.

References

1. Alon, U.: An Introduction to Systems Biology: Design Principles of Biological Circuits. Chapman & Hall (2006)
2. Sîrbu, A., Ruskin, H.J., Crane, M.: Comparison of Evolutionary Algorithms in Gene Regulatory Network Model Inference. BMC Bioinformatics 11, 59 (2010)
3. Lee, W.P., Hsiao, Y.T.: Inferring Gene Regulatory Networks Using A Hybrid GA-PSO Approach With Numerical Constraints and Network Decomposition. Information Sciences 188, 80–99 (2012)
4. Mussel, C., Hopfensitz, M., Kestler, H.A.: BoolNet-an R Package for Generation, Reconstruction and Analysis of Boolean networks. Bioinformatics 26, 1378 (2010)
5. Hsiao, Y.-T., Lee, W.-P.: Evolving Gene Regulatory Networks: A Sensitivity-Based Approach. In: Huang, D.-S., Gan, Y., Premaratne, P., Han, K. (eds.) ICIC 2011. LNCS (LNBI), vol. 6840, pp. 508–513. Springer, Heidelberg (2012)
6. Cho, K., Shin, S., Kolch, W., Wolkenhauer, O.: Experimental Design in Systems Biology, Based on Parameter Sensitivity Analysis Using A Monte Carlo Method: A Case Study for The TNFα-mediated NF-kB Signal Transduction Pathway. Simulation 79, 726–729 (2003)
7. Cao, H., Kang, L., Chen, Y.: Evolutionary Modeling of Systems of Ordinary Differential Equations with Genetic Programming. Genetic Programming and Evolvable Machines 1, 309–337 (2000)
8. Ronen, M., Rosenberg, R., Shraiman, B.I., Alon, U.: Assigning Numbers to The Arrows: Parameterizing A Gene Regulation Network by Using Accurate Expression Kinetics. PNAS 99, 10555–10560 (2002)
9. Bansal, M., Gatta, G.D., di Bernardo, D.: Inference of Gene Regulatory Networks and Compound Mode of Action from Time Course Gene Expression Profiles. Bioinformatics 22, 815–822 (2006)
10. Kimura, S., Sonoda, K., Yamane, S., Maeda, H., Matsumura, K., Hatakeyama, M.: Function Approximation Approach to The Inference of Reduced NGnet Models of Genetic Networks. BMC Bioinformatics 9, 23 (2008)
11. Kabir, M., Noman, N., Iba, H.: Reversely Engineering Gene Regulatory Network from Microarray Data Using Linear Time-variant Model. BMC Bioinformatics 11, S56 (2010)
12. Bazil, J.N., Qi, F., Beard, D.A.: A parallel algorithm for reverse engineering of biological networks. Integrative Biology 3, 1145–1145 (2011)

Credit Scoring Based on Kernel Matching Pursuit

Jianwu Li, Haizhou Wei, Chunyan Kong, Xin Hou, and Hong Li

Beijing Key Lab of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
ljw@bit.edu.cn

Abstract. Credit risk is paid more and more attention by financial institutions, and credit scoring has become an active research topic. This paper proposes a new credit scoring method based on kernel matching pursuit (KMP). KMP appends sequentially basic functions from a kernel-based dictionary to an initial empty basis using a greedy optimization algorithm, to approximate a given function, and obtain the final solution with a linear combination of chosen functions. An outstanding advantage of KMP in solving classification problems is the sparsity of its solution. Experiments based on two real data sets from UCI repository show the effectiveness and sparsity of KMP in building credit scoring model.

Keywords: Credit scoring, Kernel matching pursuit, Support vector machine.

1 Introduction

The specific duty of a credit scoring model is to assign credit applicants to either a 'good credit' group that are certain to repay financial obligation or a 'bad credit' group with high possibility to default on financial obligation. Recently, many automatic credit scoring algorithms have been proposed to build credit scoring models. Initially, some statistical and optimization techniques were widely employed such as linear discriminate analysis [1] and logistic regression [2]. Then, some more powerful methods from artificial intelligence were also used to build credit scoring models such as decision trees [3], artificial neural networks [4], genetic programming [5], cased-based reasoning [6] and support vector machines [7]. Among all these methods, decision trees, artificial neural networks and support vector machines are generally regarded as the more effective credit scoring models.

This paper proposes to apply kernel matching pursuit (KMP) to credit scoring.

2 Kernel Matching Pursuit

2.1 Kernel Matching Pursuit (KMP)

KMP, proposed by Vincent and Bengio [8], has an outstanding advantage on the sparsity of the solution. KMP originates from matching pursuit (MP) algorithms.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 118–122, 2013.

MP algorithm learns a function that is a linear combination of functions choosing from a basis function dictionary, by sequentially appending basic functions to an initial empty basis using a greedy optimization algorithm, to approximate a given function.

KMP, as a special MP algorithm, uses a kernel-based dictionary and is applied to the field of machine learning. Given a kernel function K: $R^d \times R^d \to R$, $D = \{d_i = K(\cdot, x_i) | i = 1, 2, \ldots, l\}$ is used as the dictionary of KMP [8].

Training a KMP can be decomposed into three steps. First, it builds the basis function dictionary $D = \{d_i = K(\cdot, x_i) | i = 1, 2, \ldots, l\}$, where K is a given kernel function. The second step is an iterative process, and at each iterative step, the most correlative basis function in D with the current residual error $\overrightarrow{R_n}$ and its corresponding coefficient α_n are calculated. At last, when the predefined threshold is satisfied the iterative process is stopped and the final decision function has the form,

$$f(x) = \sum_{n=1}^{N} \alpha_n K(x, x_n) \tag{1}$$

where $K(\cdot, x_n)$ is the basic function which is found in the preceding iterative process.

Clearly, the solution of KMP has a similar form to that one from SVM. The kernel function of SVM must satisfy the Mercer condition, while KMP has no restriction on the shape of the kernel. Besides, KMP can mix different kernel types in one dictionary. In this paper, the Gaussian kernel function $K(x_1, x_2) = \exp(-\|x_1 - x_2\|^2 / \sigma^2)$ was selected for all KMP models.

Several improved versions, such as the KMP with back-fitting and the KMP with pre-fitting, were also further proposed. Their details refer to [8].

2.2 Kernel Matching Pursuit for Classification

The problem on credit scoring can be considered as a binary classification issue, and the observation value y belongs to {-1, 1} which is discrete, while the prediction function $f(x)$ of KMP outputs continuous values. In order to obtain the prediction category, $m = y \cdot f(x)$ can be seen as a confidence measure and the classifying result is given by sign($f(x)$). Meanwhile, the margin loss functions used in neural networks can be adopted for KMP to deal with classification problem [8].

- Squared loss: $(\hat{f}(x) - y)^2 = (1 - m)^2$;
- Squared loss after tanh with modified target: $(\tanh(\hat{f}(x)) - 0.65y)^2 = (0.65 - \tanh(m))^2$, where $m = y\hat{f}(x)$, called individual margin at point x.

Since the squared loss penalizes large positive margin, the decision surface is drawn towards the cluster of training data at the expense of a few misclassified points [8]. However, squared loss after tanh with modified target can correct this problem. Therefore, the squared loss after tanh with modified target was used for all KMP models in the following experiments.

3 Experiments

Two real world data sets, Australian and German credit datasets, which are from UCI Machine Learning Repository [9], were used to test the performance of the proposed model. Australian credit data set consists of 307 good credit samples and 383 bad ones. German credit data set consists of 700 good credit samples and 300 bad ones. In all experiments, the programs were run based on Matlab 7.8.0.

Each dataset was randomly divided into two parts, training dataset making up 2/3 of the total dataset and testing dataset making up the remaining 1/3. Each experiment was performed for 30 independent runs, and the average result was reported. Additionally, the KMP with back-fitting was adopted [8].

For the convenience of description, the number of creditworthy cases classified as good is denoted as GG; the number of creditworthy cases classified as bad is written as GB; the number of default cases classified as good is represented as BG; the number of default cases classified as bad is expressed as BB. Three commonly used evaluation criteria measuring the effectiveness of classification are as follows,

$$Sensitivity=\frac{GG}{GG+GB},$$

$$Specificity=\frac{BB}{BB+BG},$$

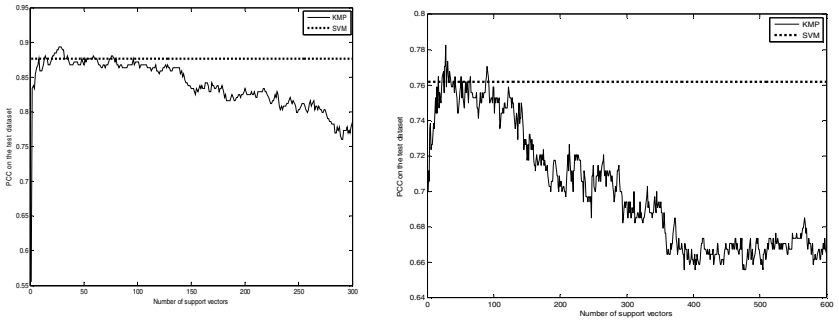
$$Percentage\ correctly\ classified(PCC)=\frac{GG+BB}{GG+GB+BB+BG}.$$

The PCC of KMP on test set was computed as a function of the number of support vectors — N. The results are showed in Fig. 1 (a) for Australian dataset and Fig. 1 (b) for German dataset. From Fig. 1, the PCC of KMP rises with the parameter N increasing before N reaches a specific value. And after the parameter N is larger than the specific value, the PCC of KMP decreases with N increasing. So the PCC of KMP model reaches the maximum when N is around the specific value. This is because when N is too small, KMP cannot learn from training dataset sufficiently, while when N is too large, KMP is over-learning. Hence, an appropriate value of the parameter N should be chosen.

In order to test the effectiveness of KMP in building credit scoring model, we compared the performance of KMP with SVM. As mentioned before, SVM has been regarded as one of the most effective credit scoring models. We selected Gaussian kernel for SVM and used grid search to find the best parameters of SVM. The PCC of SVM is also described in Fig. 1. When N is around a specific value, the PCC of KMP is larger than SVM. We also show the other evaluation criteria of KMP with several different values of N as well as those of SVM in Table 1 for Australian dataset and Table 2 for German dataset.

From Table 1 and Table 2, when the appropriate parameter N is selected, KMP has higher specificity and higher PCC than SVM, while it has lower sensitivity than SVM. This means that KMP has higher total classification accuracy and higher accuracy to detect bad credit samples than SVM. However, compared to SVM, KMP has larger probability to reject good credit applicants.

As Table 1 and Table 2 show, KMP uses only about 30 support vectors while SVM uses hundreds of support vectors. Therefore, KMP can obtain a more sparse classifier and use less training time than SVM. Totally speaking, KMP performs better than SVM.



(a) Australian dataset (b) German dataset

Fig. 1. PCC of KMP as a function of the number of support vectors. The thick dotted line represents the PCC of SVM

Table 1. Performance of KMP with different numbers of support vectors for Australian dataset and comparison with SVM

	#s.v.	Sensitivity (%)	Specificity (%)	PCC (%)	Training time (s)
	25	87.50	90.00	88.89	0.333
KMP	28	87.50	90.77	89.32	0.349
	31	88.46	89.23	88.89	0.355
SVM	207	93.27	83.08	87.61	1.723

#s.v.: number of support vectors.

Table 2. Performance of KMP with different numbers of support vectors for German dataset and comparison with SVM

	#s.v.	Sensitivity (%)	Specificity (%)	PCC (%)	Training time (s)
	25	87.39	52.94	77.06	0.695
KMP	28	87.82	55.88	78.24	0.705
	31	87.39	53.92	77.35	0.739
SVM	442	88.24	48.04	76.18	2.702

4 Conclusions and Future Research

This paper proposes to use KMP to build credit scoring model. Experimental results show that the performance of KMP is better than SVM which has been considered as

one of the most effective credit scoring models. KMP is more accurate and uses less support vectors. Besides, KMP takes less training time.

In future research, we will attempt to apply KMP models with other loss functions to credit scoring, and also to use KMP ensemble to solve the credit scoring problem with a large-scale dataset.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (No. 61271374) and the Beijing Natural Science Foundation (No. 4122068).

References

1. Myers, J.H., Forgy, E.W.: The development of numerical credit evaluation systems. Journal of the American Statistical Association 58(303), 799–806 (1963)
2. Wiginton, J.C.: A note on the comparison of logit and discriminant models of consumer credit behavior. Journal of Financial Quantitative Analysis 15(3), 757–770 (1980)
3. Zhou, X.Y., Zhang, D.F., Jiang, Y.: A new credit scoring method based on rough sets and decision tree. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 1081–1089. Springer, Heidelberg (2008)
4. West, D.: Neural network credit scoring models. Computers and Operations Research 27(11-12), 1131–1152 (2000)
5. Abdou, H.A.: Genetic programming for credit scoring: The case of Egyptian public sector banks. Expert Systems with Applications 36(9), 11402–11417 (2009)
6. Chuang, C.L., Lin, R.H.: Constructing a reassigning credit scoring model. Expert Systems with Applications 36(2), 1685–1694 (2009)
7. Zhou, L., Lai, K.K., Yu, L.: Credit scoring using support vector machines with direct search for parameters selection. Soft Comput. 13, 149–155 (2009)
8. Vincent, P., Bengio, Y.: Kernel Matching Pursuit. Machine Learning 48, 165–187 (2002)
9. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. School of Information and Computer Science, University of California(2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`

Vehicle Queue Length Measurement Based on a Modified Local Variance and LBP*

Qin Chai, Cheng Cheng**, Chunmei Liu, and Hongzhong Chen

Key Laboratory of Embedded System and Service Computing,
Ministry of Education Tongji University, 201804 Shanghai, China
{chaiskyok,chengcheng_lcc}@163.com,
chunmei.liu@tongji.edu.cn, tjchz@sina.com

Abstract. The real-time traffic parameters are necessary to dynamic traffic light control at intersection due to the serious traffic congestion. In this paper, we describe an approach for the real-time vehicle queue length measurement in a video-based traffic monitoring system. It is built on the property of a modified local variance in video frames, which does not rely on any sort of motion detection. In addition, a shadow removal approach is also presented with a simplified LBP as well as this local variance. Experimental results show that the proposed approach can highly improve the performance of the queue length measurement in real time, and it can be efficiently applied in a variety of traffic scenes.

Keywords: Modified local variance, local binary pattern, vehicle queue length.

1 Introduction

The increase of traffic congestion in urban area has made essential the use of intelligent transportation systems (ITS). ITS can provide dynamic traffic management according to the real-time scenarios. Many works on ITS aim at providing traffic parameters, which mainly rely on the technique of ground induction coil or floating car data. Compared with such traditional methods, ITS based on video analysis can provide a very low computational load and easy maintenance. In addition, it can cover a wider area to collect more visualized information, such as the vehicle queue length, which can provide the intuitive perception of traffic situation.

Vehicle queue is caused by the traffic lights or jams at intersections, and its measurement can be a great help to traffic light control in ITS. Generally, vehicle queue length measurement with video analysis includes vehicle detection and length measurement [1-2]. In vehicle detection, most approaches are built on the techniques of frame differencing, background subtraction and optical flow estimation. Frame differencing is easy to manipulate, but sensitive to the speed level of vehicles. Background subtraction mainly relies on the technique of

* This work was supported by the 2010 Innovation Action Plan of Science and Technology Commission of Shanghai Municipality under Grant 10DJ1400300 and the National Natural Science Foundation of China under Grant 61003102 and 61003221.

** Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 123–128, 2013.

background modeling, which is not applicable for the static vehicles. As for optical flow estimation, the heavy cost in computation has restricted its applications. In summary, motion information is involved more or less in such approaches. None of them can detect the static vehicles well at intersections. To solve this problem, several methods based on feature extraction have been proposed recently [3-7]. Fathy *et al.*[3-4] propose to detect the presence of cars with spatial edges on the sub-profiles. It estimates the vehicle queue length by neural networks, which has expensive cost in computation. In [5], Y. Qiao *et al*. integrate the methods of edge extraction and frame differencing to detect moving objects, which is not suitable for static vehicles. A method based on corner detection is presented in [6], and the features are classified as static or moving in measuring the queue length. Combined with an entropy-based texture description, the authors in [7] utilize the edge detection to measure the length, but it is sensitive to the road marks and noises.

In this work, a three-step procedure is proposed to accomplish the vehicle queue length measurement. First, a method of modified local variance is used for vehicle detection. Second, the cast shadow by vehicle is removed with a simplification of local binary pattern (LBP) as well as this local variance. Finally, the vehicle queue length is measured by camera calibration.

2 Measuring the Vehicle Queue Length

2.1 Vehicle Detection Based on a Modified Local Variance

In an image with traffic scenes, the grayscale of vehicles always changes significantly, while the one in the road surface keeps nearly constant. Typically, the variance in vehicles is larger than the one in background. Based on such observations, we apply the local variance as image content detector. Local variance means that it is locally computed. Formally, it can be formulated as:

$$Var(I_p) = E\{I_p^2\} - (E\{I_p\})^2 \tag{1}$$

where I_p denotes the gray value at pixel *p*, $E\{I_p\} = \sum_{q \in N_p} w_q I_q$ and $E\{I_p^2\} = \sum_{q \in N_p} w_q I_q^2$, in which w_q is the normalized weight in neighborhood N_p. In practice, N_p is chosen as a block with Gaussian weights. To further discriminate the difference between vehicles and road surface, a property of variance can be used, which states that for any constant *C*, the variance of CI_p is scaled as C^2 times of the original one. Therefore, the difference between these two variances is enlarged when $C>1$. Thus, in our approach, a modification for computing the local variance is replaced as:

$$Var(CI_p) = E\{(CI_p)^2\} - (E\{CI_p\})^2 \tag{2}$$

It must be noted that the intensity of this variance image changes frame by frame, so it is normalized first. Then, to detect the foreground of vehicles, a binary operation is adopted, and the vehicles are detected. In our experiment, it is found that the threshold selection is not quite strict. A threshold *T* around 0.9 meets the requirement. For most

cases, the result will be sensitive to the noise in background when the threshold $T<0.85$, and it will lose most vehicle information with $T>0.99$.

In practice, this modified local variance can be computed by convolution. Since the Gaussian kernel is separable in nature, hence, the computation cost is greatly reduced. In figure 1, the effect of this vehicle detection method is presented, where the threshold T is set as 0.88. It is obvious that the information of vehicles is well preserved, while the noise from the road, even as the road marks, has been discarded. Although the buildings along the roadside have also been detected, they are not interested in our purpose, since the region of interest (ROI) is just set on the lanes.

Fig. 1. Vehicle detection. The gray image (left) and its the modified local variance (right)

2.2 Shadow Removal Based on a Simplified LBP

Shadows cast by vehicles must be considered carefully in the video-based ITS. Since the shadows have the similar dynamics with vehicles, they will be misclassified as part of the foreground. In our systems, the shadows, which are cast by the vehicles in adjacent lane, may cause significant errors in the vehicle queue length measurement. In this section, we will present an approach for shadow removal, which is based on a simplified LBP.

LBP is a means of texture description and has been extended to a unified model with different scales [8]. The original LBP is computed as:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) \times 2^p, s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{3}$$

where g_p corresponds to the values of P equally spaced pixels on a circle of radius R around a central pixel g_c.To reduce the computational cost, a simplified LBP is made with just half of the pixels, while the performance is preserved. It is formulated as:

$$LBP_{P,R} = \sum_{i=1}^{P/2} s(g_{2i-1} - g_c) \times 2^{2i-1}, s(x) = \begin{cases} 1 & abs(x) \geq T \\ 0 & otherwise \end{cases} \tag{4}$$

where T is a properly chosen threshold. The effects of these two LBP-based texture descriptions are presented in figure 2.

Combined with the local variance, the procedure of shadow removal is composed of the following three steps. **Step 1:** compute the modified local variance, then a binarization with two thresholds $T1$ and $T2$ ($T1<T2$) is operated to obtain the corresponding masks A and B. Subsequently, compute the simplified LBP. **Step 2:** a

mask *C* is obtained by eroding the mask *A*, then an "and" operator is executed on mask *C* and the simplified LBP image. The output is denoted as mask *M*. **Step 3:** an "or" operator is applied to mask *M* and *B* to obtain the output of mask *N*. By morphological dilation on mask *N*, the final shadow removed vehicle is presented as the mask *S*. A flow chart for the whole vehicle shadow removal is described in figure 3.

Fig. 2. The original gray image (left), LBP image (middle), the simplified LBP image (right)

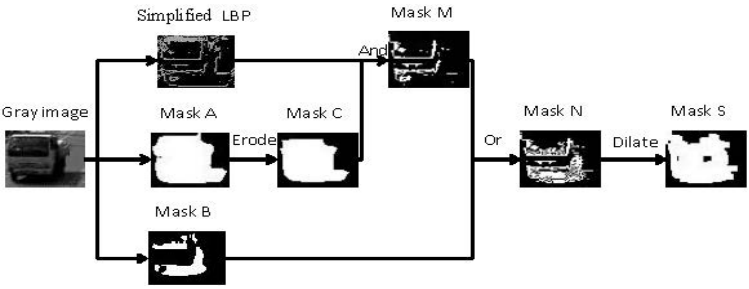


Fig. 3. Flow chart of shadow removal

2.3 Image Geometric Correction

The images captured by cameras contain the factor of perspective. They are not appropriate for length measurement. Thus, creating a visual model between the road plane and image plane is critical in our systems. Here, all the operations are processed in ROI, and the longitudinal dimension of vehicles is adopted as the queue length.

Under the assumption of a pinhole camera model, the camera coordinates can be projected onto a zenith view through a homography matrix *H*, which is determined with at least 4 pairs of points from the real world and image plane. Once the homography matrix is obtained, the image of ROI from the zenith view can be computed. Figure 4 shows the effect of such correction with a traffic video frame.

2.4 Measuring the Vehicle Queue Length

After geometric correction, the obtained zenith view image is represented with the real world coordinate systems. To measure the vehicle queue length, a three-step procedure is executed. **Step 1:** project all the foreground points in ROI onto the central axis. **Step 2:** compute the distance between the first and the last projected points on the central axis, which is defined as the length of the foreground L_1. **Step 3:**

due to the effect of perspective, the measurement error caused by blind zone cannot be ignored. The actual length L is computed by eliminating the length of blind zone l as follows $L = L_1 - (L_1 + d)h/H$, where H is the camera installation height, d denotes the distance between camera pole and zebra line, and h is a preset vehicle height on average.

Fig. 4. The geometric correction

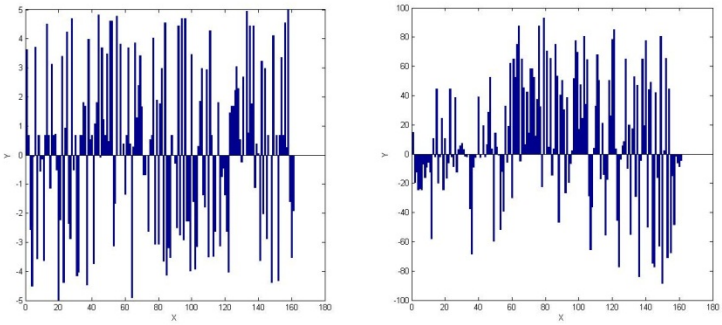


Fig. 5. Comparison between the proposed method (left) and the method in [7] (right). The X coordinate denotes the index of frames and the Y coordinate represents the discrepancy between the actual vehicle queue length and its estimation.

3 Experiments and Analysis

The proposed approach has been tested on the data set collected at an intersection in Qidong of Jiangsu province. In experiments, the video frame is with the size of 352×288, which meets the requirement in our system. In the case of foreground extraction with local variance, the block size is chosen to be 5×5, and the amplification factor C is set as 25. The two thresholds $T1$ and $T2$ for shadow removal are 0.88 and 0.95, respectively.

Comparison is made with the method in [7]. The actual vehicle queue lengths for reference are measured manually in advance, and the measurement errors of the two methods are presented in figure 5. Among the 161 tested frames, it is found that the measurement error in our method is always less than the vehicle length on average, while the measurement error with the entropy-based method is too large to be

accepted. Although different thresholds have been tried, it fails to detect the actual road surface, because the degree of difference between pixels is ignored in [7].

In our video-based traffic light control system, only the frames when the traffic light turns green need to be processed. Therefore, the computation cost of our approach is reduced, and the average processing speed is 25 frames per second by Visual studio 2008 under the computer configuration with an Intel Core 2 Duo 2-GHz central processing unit.

4 Conclusions

We have presented a method for vehicle queue length measurement based on a modified local variance and LBP. It is a critical part in our video-based dynamic traffic light control system. The proposed approach is easy to manipulate and robust to a variety of scenes. It has fulfilled the requirement in the real-time applications.

References

1. Zanin, M..., Modena, C.M...: An efficient vehicle queue detection system based on image processing. In: Proceeding of 12th International Conference on Image Analysis and Processing, pp. 232–237 (2003)
2. Iwasaki, Y.: An image processing system to measure vehicle queues and an adaptive traffic signal control by using the information of the queues. In: Proceeding of IEEE International Conference on Intelligent Transportation System, pp. 195–200 (1997)
3. Siyal, M.Y., Fathy, M.: Real-time image processing approach to measure traffic queue parameters. IEE Proc.-Vis. Image Signal Process 142(5), 297–303 (1995)
4. Siyal, M.Y., Fathy, M.: A neural-vision based approach to measure traffic queue parameters in real-time. Pattern Recognition Letters 20(8), 761–770 (1999)
5. Qiao, Y., Shi, Z.K.: Traffic parameters detection using edge and texture. Procedia Engineering 29, 3858–3862 (2012)
6. Albiol, A., Mossi, J.M.: Video-based traffic queue length estimation. In: International Conference on Computer Vision Workshops, pp. 1928–1932 (2011)
7. Cai, Y.F..., Wang, H...: Measurement of Vehicle Queue Length Based on Video Processing in Intelligent Traffic Signal Control System. In: International Conference on Measuring Technology and Mechatronics Automation, Changsha, China, pp. 615–618 (2010)
8. Ojal, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 971–987 (2002)

Applying SBL and Non-Linear Dynamics Features for Detecting Deception from Speech Signal

Yan Zhou[1,2]

[1] Department of Communication Technology, Electronic Information Engineering College, Suzhou Vocational University, Suzhou Jiangsu, 215104, China
[2] School of Electronics and Information Engineering, Soochow University, Suzhou Jiangsu, 215104, China
Zhyan@jssvc.edu.cn

Abstract. Extracting novel non-linear dynamics (NLD) feature sets and applying SBL classifier for deception detection based speech processing is the primary aim of this study. As the NLD features provide additional information regarding the dynamics and structure of deceptive speech, here, 24 NLD features that show significant correlations to deception are selected. The features have been computed partially, and represent so far unknown acoustical perceptual concepts. After a correlation-filter feature subset selection, Sparse Bayesian Learning (SBL) classification model was trained. SBL algorithm is turned out to gain a satisfactory performance for detecting deceptive speakers on the NLD feature sets. Compared with the classical model of SVM and RBFNN, the proposed model achieves high classification accuracy in detecting deception.

Keywords: deception detection, speech signal, Sparse Bayesian Learning (SBL), non-linear dynamics (NLD).

1 Introduction

Since the accurate detection of deception in human interactions has long been of interest across abroad array of contexts and disciplines, considerable work relating to deception has been undertaken in fields such as psychology, physiology, communication, and to some extent, law enforcement[1]. However, the bulk of that works had focused on gestural and facial cue to deception, and limited work has been done with the aim of developing scientifically verified automatic deception detection, and even less work has focused specifically on speech. Deception detection is an unusual problem in the speech processing domain in that humans perform very poorly at the task. Thus, while matching human performance would represent considerable success in the speech recognition, speech-to-speech translation, or even emotion detection domains [2].The presented works[3-4]represent the primary comprehensive attempt to apply a broad array of techniques from spoken language processing to the

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 129–135, 2013.

tasks of detecting deception in speech, and to identifying acoustic, lexical, prosodic and paralinguistic correlates of deception. Ekman et al.[5] reported a significant increase in pitch in deceptive speech with respect to truthful speech. Newman et al.[6] applied the Linguistic Inquiry and Word Count program to texts from five studies in various combinations. Vrij[7] likewise offers an analysis of some of the literature on individual differences in verbal cues to deception. Despite of the above achievements of deception detection using speech, the detection rate of deception is still in the primary stage, so it has a very extensively research space. From the previous work in deception detection, it is noted that all the existing detection algorithms are not sparse representation processing methods. Moreover, they also do not consider the temporal correlation characteristics of deceptive speech. Therefore, their performance degrades significantly with the correlation.

The goal of this work is to extract the temporal characteristics of the deceptive speech and examine the efficacy of applying Sparse Bayesian Learning techniques to the problem of deception detecting. Because the references [8] proved that the speech signal is actually has the chaos features, actually, it can be influenced by the changes of psychological state, and physiological status. This provides a solid theoretical justification for the methodology of adapting the NLD characteristics, which are used for the SBL deception detection model. Here, from a Bayesian evidence perspective, a simplified derivation for the detection paradigm is provided. Furthermore, the detection model is applied to capture the temporal correlation characteristics, and then the problem of deception detection can be realized. In particular, this method has not previously been employed to the domain of deception detection. Most importantly, this paper has sought to demonstrate the efficacy of the proposed method through statistical analyses and classification experiments using a large number of samples. The experimental results show that such techniques could provide insights about deceptive speech behavior, and in the best case, could be employed to classify deceptive speech for the realms of business, politics, jurisprudence, law enforcement, and national security.

2 Extracting of Chaotic Time Series Feature Sets

An important factor to choose the NLD feature is that, they might supply additional information and complement traditional time and frequency domain analyses of speech. The deception influenced speech production is the generation of non-linear aerodynamic phenomena within the vocal tract. The deceptive speech includes non-laminar flow, flow separation in various regions, generation and propagation of vortices and formation of jets rather than well-behaved laminar flow. Thus, it obviously shows that the deceptive speech production is a chaos processing. The Reconstructed Phase Space (RPS) is an important technology for the research of extraction of chaotic time series features. Then, the Reconstructing

attractor can be yield, which is normally decided by three chaotic characteristics, respectively for Fractal features, Lyapunov exponents and the Kolmogorov entropy.

3 Deception Detection based on SBL Model

SBL arises from a probabilistic perspective, and it is presented as an approximation of the posterior distribution of all unknowns given the data. In this paper, the deceptive speech signals with noisy is considered. The noisy is assumed as $n \sim N(0, \sigma^2 I_M)$, here, σ^2 denotes the variance of the unknown noise, the deceptive speech signal can be sparse represented as this:

$$x = \theta \cdot w \tag{1}$$

Herein, we derive the SBL cost function as an exact evaluation of the Bayesian evidence. The likelihood function for the weighting w and the noisy variance σ^2 can be described as:

$$p(x|w,\sigma^2) = (2p\sigma)^{-\frac{M}{2}} \exp\{-\frac{\|x-\theta w\|_2^2}{2\sigma^2}\} \tag{2}$$

The general method to solve the optimal w is to use maximum likelihood, but this method often yields over-learning phenomenon. Therefore, to overcome this disadvantage, in SBL algorithm, the weighting w is assigned prior condition probability distribution at the first hand to control the complexity of the model. It can be expressed as:

$$p(w|\alpha) = \prod_{i=1}^{M} N(w_i|0, \alpha_i^{-1}) \tag{3}$$

Where, $N(w_i/0, \alpha_i^{-1})$ is the Gaussian probability density function with the Mean is zero, and the variance is α_i^{-1}, α is given Prior probability as following:

$$p(\alpha|a,b) = \prod_{i=1}^{M} \Gamma(\alpha_i|a,b) \tag{4}$$

Where, a, b are two important parameters in the Γ distribution, furthermore, the α, σ^2 are defined as super parameters. From the formula above, every component of the weighting w_i is controlled by a super parameter, and the finally Prior probability can be described as:

$$p(w|a,b)=\prod_{i=1}^{M}\int_{0}^{\infty}N(w_i|0,\alpha_i^{-1})\Gamma(\alpha_i|a,b)d\alpha_i \tag{5}$$

With SBL, given the Gaussian weight priors from (5), the posterior density of the weights is Gaussian,

$$p(w|x,\alpha,\sigma^2)\propto N(\mu,\Sigma) \tag{6}$$

With $\mu=\sigma^{-2}\Sigma\theta^T x$, $\Sigma=(A+\sigma^{-2}\theta^T\theta)^{-1}$, and $A=diag(\alpha_1,\alpha_2,...,\alpha_M)$, Thus, the onus remains in estimating α and σ^2 .Namely, The problem is converted into the solving of Logarithmic Edge likelihood function, $L(\alpha,\sigma^2)$ is expressed as following which adopts Laplace method.

$$\begin{aligned}L(\alpha,\sigma^2)&=\log p(y|\alpha,\sigma^2)=\log\int p(x|w,\sigma^2)p(w|\alpha)dw\\&=\frac{1}{2}[K\log 2p+\log|C|+y^TC^{-1}y]\end{aligned} \tag{7}$$

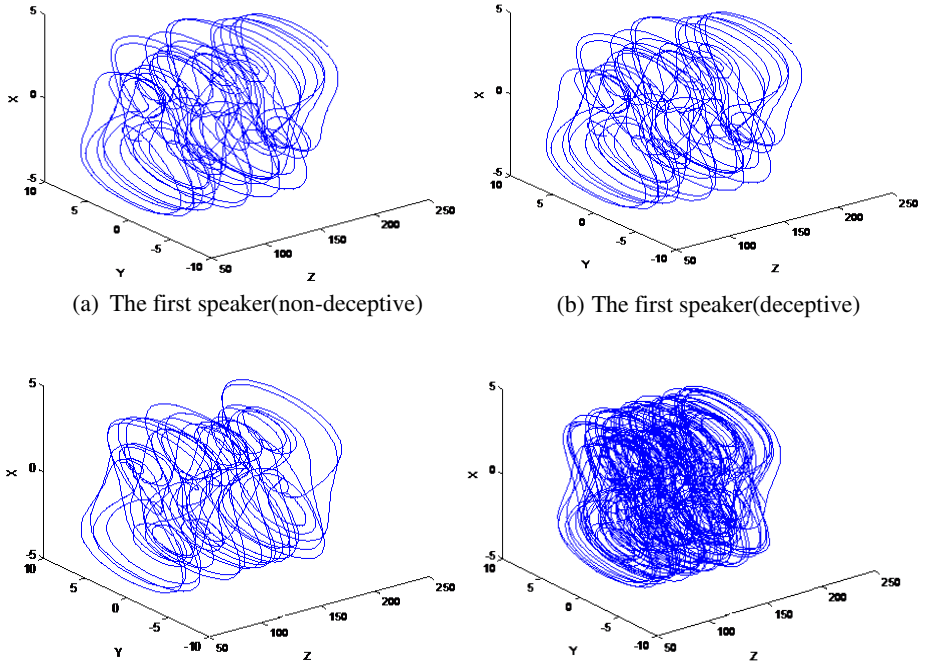
Here, $C=\sigma^2 I+\theta A^{-1}\theta^T$, to accomplish this, we employ the EM algorithm to solve the above formula until convergence.

4 Experimental Results

In this section, in order to study fractal feature and SBL effects in deceptive speech detection, experiments were performed on the professional deceptive speech corpus. The University of Arizona deceptive corpus is exploited. In our experiment, the feature sets were selected across 20 speakers from the corpus of deceptive speech. Afterwards, the presented results are compared with the classification performance.

4.1 Comparison of the Feature Effectiveness

In order to study speaker-dependent feature effects in deceptive speech, in this sub-section, all experiments were performed separately on each speaker to compare the different effectiveness. RPS figures provide a first insight into possible NLD features sensitive to deception. Therefore, RPS figures of three speakers are shown in Fig.1



(a) The first speaker(non-deceptive) (b) The first speaker(deceptive)

(c) The second speaker(non-deceptive) (d) The second speaker(deceptive)

Fig. 1. Reconstructed phase spaces for non-deceptive speaker (left) and deceptive speaker (right), speech sample $N = 125$, and $d = 3$, $\tau = 1, 3$

As it can be inferred from the distances between the trajectories, the non-deceptive speech attractor figures are less blurred than the deceptive ones for different speakers.

4.2 Comparison of the Detection Effectiveness

In order to validate the Classification results based on SBL model, here, the experiment for detection is prepared. Each speaker's sample was partitioned into 70% for training and 30% for testing. Moreover, the method of Support Vector Machine (SVM), and RBF neural network (RBFNN) were introduced for contrasting. For the SVM and the SBL algorithm, the Gaussian Radial Basis Kernel Function is employed. Table1 shows the classification results of three different methods with two step sizes ($\tau = 1, \tau = 3$).

From Table1, we can infer that the three methods can obtain satisfactory result, but when decreasing the number of samples, the error of RBFNN become increasing. However, the error of SBL is increasing un-conspicuous. This suggests that the algorithm of SBL has the ability of generalization.

Table 1. The comparison of MSE for three methods with different samples and step sizes

Samples	Step size	100	80	60	40	20
RBFNN	$\tau = 1$	0.0421	0.0532	0.1509	1.6278	1.8698
	$\tau = 3$	0.0574	0.0812	0.1834	1.3467	1.8974
SVM	$\tau = 1$	0.0325	0.0423	0.1374	0.8386	1.0934
	$\tau = 3$	0.0432	0.0410	0.1983	0.8751	1.3758
SBL	$\tau = 1$	0.0267	0.0376	0.0894	0.1653	0.1976
	$\tau = 3$	0.0329	0.0421	0.1037	0.2843	0.2932

5 Conclusions

The contribution of this study was to extract NLD features and apply the SBL classification model for speech based deception detection. Advantage of this deception detection approach is that in many application settings obtaining speech data is objective and non-obtrusive. Furthermore, this method allows for measurements over a period of time and adequately considered the temporal correlation characteristics of the deceptive speech. Moreover, due to the chaos characteristic is not sensitive to noisy, so the proposed feature extraction model had the performance of resistance to noise. As could be observed from the experiments, large performance differences existed between classifiers. However, the advantage of SBL classifier is that it needs less kernel functions and the condition is not strictly, which lead to fast calculation speed and small memory. Therefore, the proposed model could obtain better detection rate over SVM and RBFNN methods, and might be good enough to warrant future research. Nevertheless, little empirical research has been done to examine these relationships between deception and deceptive speech production. Thus, further research on NLD features related deception might be a promising challenge. Last but not least, it would seem significantly to enrich the detection model to explore excellent performance.

Acknowledgments. This work was supported by the National Natural Science Foundations of China (Grant No.61071215), the National Natural Science Foundation of China (Grant No. 60970058), the Applied basic research project of Suzhou city (SYG201033), Qing Lan Project of Jiangsu province, and the Innovative team foundation of Soochow vocational university.

References

[1] Bond Jr., C.F., DePaulo, B.M.: Accuracy of deception judgments. J. Personality and Social Psychology Review 10, 214–234 (2006)

[2] Vogel, A.P., Fletcher, S., Maruff, P.: Acoustic analysis of the ehhects of sustained wakefulness on speech. J. Acous. Soc. Am. 128(6), 3747–3756 (2010)

[3] Bond Jr, C.F., DePaulo, B.M.: Accuracy of deception judgments. J. Personality and Social Psychology Review 10(3), 214–234 (2006)
[4] Huang, D.S., Du, J.-X.: A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. J. IEEE Transactions on Neural Networks 19(12), 2099–2115 (2008)
[5] Ekman, P., Sullivan, M., Friesen, W., Scherer, K.: Face, voice, and body in detecting deceit. J. Journal of Nonverbal Behaviour 15(2), 125–135 (1991)
[6] Newman, M.L., Pennebaker, J.W., Berry, D.S., Richards, J.M.: Lying words: Predicting deception from linguistic style. J. Personality and Social Psych. Bull. 29, 665–675 (2003)
[7] Vrij, A.: Detecting Lies and Deceit: Pitfalls and Opportunities, 2nd edn. The Psychology of Crime, Policing and Law. John Wiley & Sons, Ltd., West Sussex (2008)
[8] Zheng, C., Zhang, L., Vincent, T.-Y.N..., Huang, D.S.: Metasample-based sparse representation for tumor classification. J. IEEE/ACM Transactions on Computational Biology and Bioinformatics 8(5), 1273–1282 (2011)

Face Recognition Based on Random Weights Network and Quasi Singular Value Decomposition

Zhenghua Zhou, Jianwei Zhao, and Feilong Cao*

Department of Mathematics, China Jiliang University, Hangzhou 310018, China
zzhzjw2003@163.com, zhaojw@amss.ac.cn, feilongcao@gmail.com

Abstract. This paper proposes a novel approach of feature extraction called quasi singular values decomposition (QSVD), which can be used to obtain the algebraic features of the original images. An effective classifier, named random weights network (RWN), is applied to improve the learning speed. Integrating QSVD with RWN, fast discrete curvelet transform (FDCT), and 2-dimensional principal component analysis (2DPCA), a new method for face recognition is designed. The experimental results illustrate that the proposed method has an outstanding superiority in the aspects of separability and recognition rate.

Keywords: Face recognition, Quasi singular value decomposition, Random weights network, Fast discrete curvelet transform, 2DPCA.

1 Introduction

It is well known that feature extraction and classification are two key steps in the process of face recognition. The goal of feature extraction is to give an effective representation of facial images, while the task of classification is to distinguish those extracted features with a good classifier. So an effective face recognition system depends greatly on the effective representation of human face feature and the design of classifier. Usually, image features can be divided into four classes: visual features, statistical pixel features, transform coefficient features (i.e. geometric feature), and algebraic features, where the latter two are often used in face recognition system. At the early stage, wavelet transform is popular and widely applied in face recognition system for its multi-resolution character. Although wavelet is suitable for detecting point singularities in image, it usually fails to represent curved discontinuities. Recently, another transform method, called curvelets transform, was proposed by Donoho and Duncan [2], and it has been widely used in face recognition (see [7]). The main reason for choosing curvelet transform is that curvelet transform can improve the directional capability, that is, it can be used well to represent the edges that often have singularities. In other word, curvelet transform can well describe the edge information and better represent the curve information for the images.

On the other hand, the algebraic features of images can always well reflect intrinsic properties of images stably. The algebraic features have been considered as valid

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 136–141, 2013.

features for face recognition in [5]. As one of effective algebraic features method, singular value decomposition (SVD) [3] based recognition was applied to extract feature vectors in face recognition system [9]. SVD based feature extraction can well represent the algebraic features from image space domain. However, the processes of extracting feature vector with SVD are independent, that is, SVD is used for each image separately. While images usually come from the same class in practical applications, that is, they are often dependent for each other. Obviously, it is not appropriate to just use SVD in that case. In this paper, we will propose a new method, called quasi single value decomposition (QSVD) method, to consider the dependence of each image when we extract feature vector for each image.

In addition, we note that almost all relative literatures only use one of above features method in the process of extracting face features. Here we will use the transform coefficient features and the algebraic features simultaneously to improve the recognition rate.

It is well known that some traditional training approaches for feedforward neural networks (FNNs), such as perceptron and backpropagation (BP) algorithm, often face difficulties in tuning parameters manually, which results in the slow learning speed. If the hidden weights and biases are chosen randomly, i.e., they are considered as random variables obeying the uniform distribution on (0,1), then the least square model of linear equation with hidden layer output matrix can be used to estimate the output weights of FNNs by calculating the Moore-Penrose generalized inverse. This idea can be found in the paper [8], and was named as so-called extreme learning machine (ELM) in [4]. In fact, this idea is an algorithm on random weights network (RWN), which has a faster learning speed than the traditional BP algorithm. Unlike traditional BP algorithm and its various modified algorithms, RWN has concise architecture and no need to tune input weights and biases. Up to now, RWN has been widely applied in many areas [6, 10]. In this paper, we will select RWN as a classifier for face recognition.

The rest of this paper is organized as follows. Section 2 proposes a novel method for face recognition that is composed of FDCT, 2DPCA, the proposed QSVD, and RWN classifier. Experimental comparisons of our proposed method with some other state-of-the-art approaches for face recognition are given in Section 3. At last, conclusions based on the study are highlighted in Section 4.

2 Proposed Face Recognition Method

2.1 Feature Extraction with FDCT

In our work, curvelet transform [1] is employed to generate initial feature matrices to represent human face. In our proposed method, we will extract features from the original facial images with the FDCT proposed in [1]. We use $\hat{f}(n_1, n_2)$ to denote discrete Fourier transform of the function $f(n_1, n_2)$. Let $U_j(\omega)$ be the localizing window, and $\hat{U}[n_1, n_2]$ be supported on some rectangle with the length L_{1j} and the width L_{2j}. Now the FDCT via wrapping can be summarized as follows [1]: (1) For a

given bivariate function f, calculate its coefficients with 2-dimensional fast Fourier transform (2D-FFT) to get its representative samples $\hat{f}(n_1, n_2)$ on the Fourier frequency; (2) At each scale *j* and angle *l*, form the product $\tilde{U}_{j,l}(n_1, n_2)\hat{f}(n_1, n_2)$; (3) Wrap this product around the origin and obtain $\hat{f}_{j,l}(n_1, n_2) = W(\bar{U}_{j,l}\hat{f})(n_1, n_2)$, where the ranges of n_1, n_2 and θ are $0 < n_1 < L_{1,j}, 0 < n_2 < L_{2,j}$, and $(-\pi/4, \pi/4)$, respectively; (4) Finally, apply inverse 2D FFT to each $\hat{f}_{j,l}$ and save discrete curvelet coefficients.

2.2 Dimensionality Reduction with 2DPCA

Since the dimension of coarse curvelet coefficients matrix is still high, we need to perform the work of dimensionality reduction for the face recognition system. Here we use 2DPCA [11] to reduce the dimension of the extracted features. In general, there are more than one optimal projection axes. We usually select a set of optimal projection axes $\{\mathbf{X}_1, \ldots, \mathbf{X}_d\}$, which are subjected to the orthonormal constraints and can maximize generalized total scattered criterion. In fact, the optimal projection axes are the orthonormal eigenvectors of the coarse coefficients covariance matrix $\mathbf{C}$ corresponding to d largest eigenvalues. Now for each coarse subband coefficient matrix $\mathbf{A}$, we can compute the principal component of the matrix $\mathbf{A}$ as follows: $\mathbf{Y}_i = \mathbf{A}\mathbf{X}_i$. Those principal component vectors form an $m \times d$ matrix $\mathbf{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_d\}$, which is called feature matrix. In this way, each facial image matrix can be turned into a feature matrix.

2.3 Feature Extraction with the Proposed QSVD

Generally, original face images often contain abundant geometric and algebraic information. While the main effect of feature extraction with FDCT and dimensionality reduction with 2DPCA is to capture the part of geometric information. In order to acquire the part of algebraic information, the existing methods often use the SVD on each original image matrix to get its corresponding singular values that form a feature vector.

SVD provides an effective method for extracting feature vector from each image $\mathbf{A}_i$, where the processes of extracting feature vector $\mathbf{S}_i$ are independent. However, images usually come from the same class in practical applications, that is, they are often dependent on each other. Obviously, it is not appropriate to just use SVD in that case. Here, we propose the QSVD method that considers the dependence of each image to extract feature vector for each image. Compared with the method just using SVD, our proposed method seems more logical. It can be stated as follows.

2.3.1 Proposed QSVD Method

(1) For a set of training images $\{\mathbf{A}_i\}_{i=1}^{N_1} \in \mathbb{R}^{m\times n}$, compute their mean matrix $\overline{\mathbf{A}}$; (2) Calculate the corresponding orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ for $\overline{\mathbf{A}}$ with SVD: $\overline{\mathbf{A}} = \mathbf{U}\sum_{\overline{\mathbf{A}}}\mathbf{V}^T$;(3) For each image $\mathbf{A}_i$, calculate the matrix $\mathbf{B}_i = \mathbf{U}\mathbf{A}_i\mathbf{V}^T$ with the obtained orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$; (4) Let $p = \min\{m,n\}$, choose p largest diagonal elements of the matrix $\mathbf{B}_i$ to form the feature vector $\mathbf{S}_i$ for the image $\mathbf{A}_i (i = 1,\ldots,N_1)$.Finally, connecting the geometric feature vectors obtained by FDCT and 2DPCA with the algebraic feature vectors acquired by QSVD, we can get the final compound feature vectors.

2.4 Classification with RWN

After the final compound feature vectors are obtained, we begin to perform recognition task with the RWN classifier. Compared to some other classifiers, RWN can overcome the slow learning speed. The training of FNNs with single-hidden layer is turned into a linear system via randomly chosen hidden weights. Then the output weights can be conveniently calculated through a simple generalized inverse operation of the hidden layer output matrix.

Given a set of sample data $\{(\mathbf{x}_i, t_i)\}_{i=1}^{N} \subseteq \mathbb{R}^n \times \mathbb{R}^m$, the output of an FNN with L hidden nodes can be written as $f(x) = \sum_{i=1}^{L}\beta_i G(a_i, b_i, x)$, $\mathbf{x} \in \mathbb{R}^n$, where $\mathbf{a}_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ are the hidden weights, $\beta_i \in \mathbb{R}^m$ is the output weight connecting the *i*-th hidden node to the output node, and $G(\mathbf{a}_i, b_i, \mathbf{x})$ $(i = 1,\ldots,L)$ is the output of the *i*-th hidden node with respect to the input vector $\mathbf{x}$ For a given set of sample data $\{(\mathbf{x}_i, t_i)\}_{i=1}^{N} \subseteq \mathbb{R}^n \times \mathbb{R}^m$, we choose an FNN with L hidden-layer nodes, where the hidden-node output function $G(\mathbf{a}, b, \mathbf{x})$ and the hidden-node number L are chosen in prior. The detailed implementation is as follows:(1) Randomly assign hidden-layer weights $(\mathbf{a}_i^*, b_i^*)$; (2) Compute the hidden-layer output matrix $\mathbf{H}$ [4]; (3) Calculate the output weight vector $\hat{\beta} = H^+T$, where $\mathbf{H}^+$ is the Moore-Penrose pseudo inverse of $\mathbf{H}$, and $T = [t_1, t_2, \ldots, t_N]^T$.

3 Experiments

In our experiments, for every face database mentioned in this paper, we select randomly a part of images as training samples and the remaining as testing samples. Here, we choose about 50% of each individual images as training sampling and the rest as testing samples. All the training and testing images are decomposed with FDCT at three scales and eight different angles, and 25 subband coefficient matrices are acquired. All the experiments are carried out in MATLAB R2010a environment running on a desktop with CPU AMD Athlon 2.7GHZ and 1.75 GB RAM.

Experiment 1. In this experiment, we compare our proposed method with some state-of-the-art approaches on Yale Database. Table 1 gives the corresponding experimental results based on 5 principal components and 200 hidden nodes for Yale Database.

Table 1. Comparison of recognition rate (%) for Yale Database

Methods	Recognition rate(%)
stand eigenface[15]	76
waveletface[6]	83.3
waveletface+LDA[3]	83.5
waveletface+weighted modular PCA [31]	83.6
waveletface+KAM[27]	76
curveletface+PCA+LDA[19]	83.3
our proposed	84

As observed from Table 1, the results of our proposed method outperform some other existing face recognition methods. Since the curveletface+ PCA+LDA method [7] has better performance than some other existing methods except our proposed method, we have only compared our proposed method with it in the remaining experiments for simplicity.

Experiment 2. In this experiment, we compare the recognition rate of our proposed method with the curvelet based PCA+LDA method on Sheffield and JAFFE Databases, respectively, where we use 5 varying number of principal components with 2000 and 1200 hidden nodes, respectively. Furthermore, we also carry out the similar experiments on ORL and Face94 Databases with 4000 and 5000 hidden nodes, respectively. The corresponding experimental results are shown in Table 2, respectively.

Table 2. Average recognition rates for Sheffield and JAFFE Databases with different numbers of principal components

	Sheffield		JAFFE	
Component	PCA+LDA	Proposed	PCA+LDA	Proposed
5	93.89	99.5	93.94	99.5
10	96.11	99.5	94.62	100
15	97.78	99.5	94.84	100
20	99.44	99.75	96.58	100

4 Conclusions

This paper proposed a novel method for face recognition, which is a combination of FDCT, 2DPCA, proposed QSVD and RWN classifier. As a main highlight of our method, a novel feature extraction, called QSVD, was proposed to obtain the algebraic features of the original image, which is integrated with geometric features acquired by FDCT and 2DPCA to obtain final feature vectors for the original images. Such features

contain abundant geometric and algebraic information, which can enhance separability of images and improve recognition rate to some extent. In addition, in order to improve recognition speed, RWN, as a classifier, was used in our proposed recognition system. Comparison experiments of the proposed method with some other state-of-the-art methods for face recognition have been performed on five well-known face databases, and the experimental results showed that the proposed method can achieve higher recognition rate and take less training time.

Acknowledgments. This research was supported by the National Nature Science Foundation of China (Nos. 61101240, 61272023) and the Science Foundation of Zhejiang Education Office (No. Y201122002).

References

1. Candès, E., Demanet, L., Donoho, D., Ying, L.: Fast Discrete Curvelet Transforms. Multiscale Model. Simul. 5(3), 861–899 (2006)
2. Donoho, D.L., Duncan, M.R.: Digital Curvelet Transform: Strategy, Implementation and Experiments. In: Proceedings of SPIE, vol. 4056, pp. 12–30 (2000)
3. Horn, R.A., Johnson, C.R.: Matrix Analysis, pp. 411–455. Cambridge University Press, Cambridge (1990)
4. Huang, G.B., Zhu, Q., Siew, C.: Extreme Learning Machine: Theory and Applications. Neurocomputing 70(1-3), 489–501 (2006)
5. Hong, Z.: Algebraic Feature Extraction of Image for Recognition. Pattern Recognition 24, 211–219 (1991)
6. Mohammed, A.A., Minhas, R., Wu, Q.M.J., Sid-Ahme, M.A.: Human Face Recognition Based on Multidimensional PCA and Extreme Learning Machine. Pattern Recognition 44, 2588–2597 (2011)
7. Mandal, T., Wu, Q.M.J., Yuan, Y.: Curvelet Based Face Recognition via Dimension Reduction. Signal Processing 89(3), 2345–2353 (2009)
8. Mc Loone, S., Irwin, G.: Improving Neural Network Training Solutions Using Regularization. Neurocomputing 37, 71–90 (2001)
9. Wang, Y.H., Tan, T.N., Zhu, Y.: Face Identification Based on Singular Values Decomposition and Data Fusion. Chinese J. Comput. 23(6), 23–26 (2000)
10. Zhao, J.W., Wang, Z.H., Park, D.S.: Online Sequential Extreme Learning Machine with Forgetting Mechanism. Neurocomputing 87, 79–89 (2012)
11. Zhang, D., Zhou, Z.H.: 2DTPCA: Two-directional Two-dimensional PCA for Efficient Face Representation and Recognition. Neurocomputing 69, 224–231 (2005)

Learning KPCA for Face Recognition

Wangli Hao, Jianwu Li, and Xiao Zhang

Beijing Key Lab of Intelligent Information Technology,
School of Computer Science and Technology, Beijing Institute of Technology,
Beijing 100081, China

Abstract. Kernel principal component analysis (KPCA) is an effective method for face recognition. However, the expression of its final solution needs to take advantage of all training examples, such that its run in real-world application with large scale training set is time-consuming. This paper proposes to apply radial basis function neural network (RBFNN) to learn the feature extraction process of KPCA in order to improve the running efficiency of KPCA-based face recognition system. Experimental results based on two different face benchmark data sets, including ORL and UMIST, show that the proposed method can approach to the recognition accuracy of the original KPCA, but have sparser solutions. The proposed method can be applied to real-time or online face recognition systems.

Keywords: Kernel principal component analysis, Radial basis function neural network, Face recognition.

1 Introduction

Face recognition [1,2] has attracted enormous interests in pattern recognition and computer vision, due to its broad applications in security. Two key issues in face recognition are feature extraction and operation reduction respectively in high-dimensional space. In recent years, many subspace-based techniques for reducing data dimensions have been introduced into face recognition, such as Principal Component Analysis (PCA) [3]. PCA is good at extracting the image pixel relationship between the second order statistics, but fails in extracting the higher-order information of image. Then, the KPCA [4] is proposed to solve this problem. It firstly maps the original samples into a higher dimensional feature space, and then PCA is performed in the feature space. However, the fatal limitation of KPCA is that its computational cost is very high. The main reason lies in that the final solution obtained by KPCA depends on all training examples. To solve this problem, some methods have been introduced to improve the efficiency of KPCA. For example, the scale of training set is shrunk before feature extractors are reconstructed [5]. Consequently, the cost of the kernel matrix construction is reduced.

In face recognition, RBFNN is often utilized as classifier [6] and it shows remarkable advantages such as compact topology, fast learning speed, and optimal

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 142–146, 2013.

universal approximation ability. Different from conventional applications of RBFNN to face recognition, this paper proposes to use RBFNN to learn the feature extraction process of KPCA, aiming to improve the efficiency of KPCA-based face recognition.

2 KPCA-Based Face Recognition

KPCA has been applied to face recognition successfully because it can extract nonlinear features such as face curves effectively. It first maps nonlinearly input data into a higher-dimension feature space R^F, and then linear PCA is performed in the feature space.

The process of face recognition based on KPCA is presented as follows.

- Rewrite face image I (M, N) as a column vector I (M*N, 1), where M and N are the numbers of rows and columns of I, respectively.
- Construct the centralized kernel matrix of training examples.
- Calculate eigenvalues and eigenvectors of the kernel matrix.
- Acquire feature extractors by collecting eigenvectors corresponding to the first several largest eigenvalues in R^F.
- Obtain nonlinear features of training and test examples by mapping them onto the feature extractors, respectively, in R^F.
- Use a classifier such as nearest-neighbor algorithm, to classify test faces.

3 Learning KPCA via RBFNN for Face Recognition

The basic learning model of RBFNN is supervised and it contains three layers—input, hidden and output layers.

Although KPCA is effective in extracting nonlinear features of input data, its test process is time-consuming because the inner products of the input test example and every training example in feature space need to be computed. This seriously affects the application of KPCA-based face recognition to real-time cases. On the other hand, the number of neurons in hidden layer of RBFNN can be changed or adjusted when a RBFNN is trained. For a given approximation problem, the less the number of neurons in hidden layer, the faster the test speed of RBFNN. Thus, a natural idea is to train a RBFNN to approximate KPCA and then use the approximate model to extract face features nonlinearly. In other words, we can learn KPCA via RBFNN for face recognition.

The process of the proposed method is concretely described as follows.

Step 1. Transform training face set $I_{trainset}$ to $I_{trainfeature}$ via KPCA.
Step 2. Train a RBFNN using the input-output pairs ($I_{trainset}$, $I_{trainfeature}$).
Step 3. Transform test face set $I_{testset}$ to $I_{testfeature}$ via the trained RBFNN.
Step 4. Use a classifier such as nearest-neighbor algorithm to classify $I_{testfeature}$.

4 Experiments

To testify the effectiveness of the proposed method, the traditional KPCA for face recognition was utilized as a comparison. Two standard face benchmark data sets corresponding to ORL and UMIST, respectively, were used. Our proposed method is simply denoted as KRBFNN. The nearest-neighbor classifier was used for KPCA and KRBFNN, respectively, to realize face recognition. Additionally, every experiment was performed repeatedly 20 times and the average result was recorded.

For a given face database, assume that it contains *pnum* people, and each people has *fnum* different images where *train_face* images are training examples and the rest *test_face* images are test examples.

The parameters of the two used data sets were set as follows. For ORL, *pnum* =40, *fnum*=10, *train_face*=5 and *test_face*=5. For UMIST, *pnum* =14, *fnum*=25, *train_face*=5 and *test_face*=20. Moreover, Gaussian radial basis function was used in RBFNN.

For KPCA, the selection of its kernel function is one of the most crucial problems. Here, three usual and important kernel functions are utilized respectively including

- Gaussian kernel function, $K(x, y) = \exp(-(x-y)^2 / \sigma^2)$.
- Perceptron kernel function, $K(x, y) = \tanh(c * x * y + d)$.
- Polynomial kernel function, $K(x, y) = (x * y + a)^b$.

Table 1. The comparison of KRBFNN and KPCA based on ORL dataset

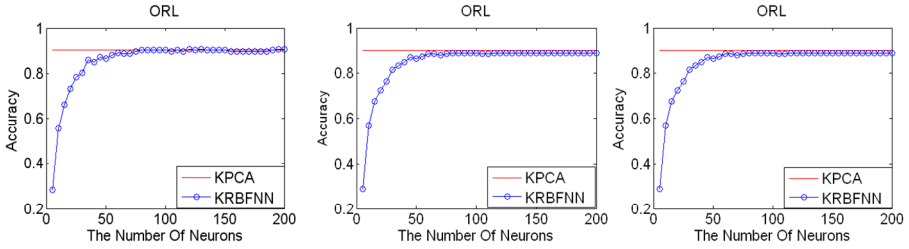
	KRBFNN				KPCA
Neurons / Kernel	50	100	150	200	Accuracy
Gaussian	0.865	0.900	0.900	0.901	0.900
Perceptron	0.865	0.900	0.900	0.890	0.880
Polynomial	0.865	0.890	0.890	0.890	0.880

Table 2. The comparison of KRBFNN and KPCA based on UMIST dataset

	KRBFNN				KPCA
Neurons / Kernel	20	40	60	70	Accuracy
Gaussian	0.918	0.936	0.936	0.939	0.939
Perceptron	0.925	0.932	0.936	0.936	0.939
Polynomial	0.925	0.932	0.936	0.936	0.939

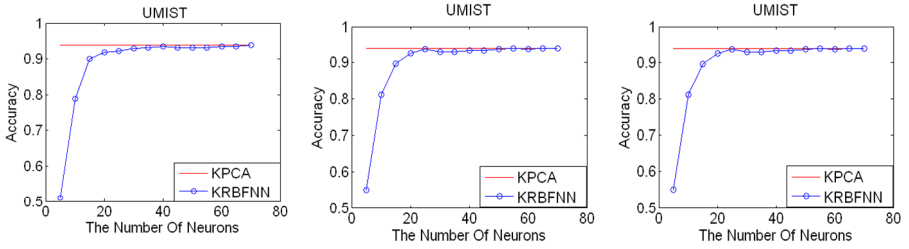
Table 1 and Table 2 show the face recognition accuracies of two methods based on ORL and UMIST datasets, respectively. When the number of neurons of KRBFNN is much less than a half of the number of training samples, the recognition accuracy of KRBFNN is almost the same as that of the traditional KPCA method.

Fig. 1 (for ORL) and Fig. 2 (for UMIST) demonstrate the approximation tendency of KRBFNN to KPCA with the number of its hidden neurons increasing. Each figure contains three subfigures and each subfigure shows the performance of two methods based on one sort of kernel functions. From left to right, the kernel functions used in three subfigures are Gaussian, Perceptron and Polynomial, respectively. KRBFNN can still acquire comparable accuracies with the original KPCA when the number of its hidden-layer neurons is much less than the number of training samples. Additionally, it should be noted that KPCA needs to use all training examples to express the final solution. Thus, we can evaluate approximately their test efficiency, according to the numbers of the used vectors in their final solutions.



(a) Gaussian kernel (b) Perceptron kernel (c) Polynomial kernel

Fig. 1. The performance of two methods for ORL



(a) Gaussian kernel (b) Perceptron kernel (c) Polynomial kernel

Fig. 2. The performance of two methods for UMIST

The reasons why RBFNN can effectively learn KPCA can be concluded as follows. First, the final solution from KPCA is related to all training examples and it is usually redundant. As some references show [5], it can be further reduced. This provides the possibility of using RBFNN to approximate it. Moreover, RBFNN is good at handling complex nonlinear approximation problems, such that the effect of learning KPCA via RBFNN can be ensured. Furthermore, according to statistical learning theory, under

the condition with comparable training accuracies, the more simple the training model, the better its generalization ability [7]. Therefore, given a certain training accuracy, we adjust the RBFNN model to make it as simple as possible. Thus, the generalization accuracy of the RBFNN can be further improved.

5 Conclusions

This paper proposes to apply RBFNN to learn KPCA for face recognition. Experimental results show that RBFNN can approximate KPCA well with much sparser basis but comparable test accuracy.

The future work is to use RBFNN to learn other kernel-based subspace face recognition systems, such as Kernel Linear Discriminant Analysis (KLDA) or Kernel Independent Component Analysis (KICA). Also, the proposed method can be applied to other kernel-based complex pattern recognition problems. Moreover, this paper provides a general framework to learn any complex nonlinear feature extraction process via RBFNN.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (No. 61271374, 61273273) and the Beijing Natural Science Foundation (No. 4122068).

References

1. Zhao, W., Chellappa, R., Phillips, P., Rosenfeld, A.: Face recognition: A literature survey. ACM Computing Surveys 35(4), 399–458 (2003)
2. Zhu, N., Li, S.: A Kernel-based sparse representation method for face recognition. Neural Computing and Applications, 1–8 (2012)
3. Kirby, M., Sirovich, L.: Application of the KL procedure for the characterization of human faces. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(1), 103–108 (1990)
4. Schölkopf, B., Smola, A., Muller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319 (1998)
5. Xu, Y..., Li, M.: A method for speeding up feature extraction based on KPCA. Neurocomputing 70(4-6), 1056–1061 (2007)
6. Er, M.J..., Toh, H.L.: Face recognition with radial basis function (RBF) neural networks. IEEE Transactions on Neural Network 13(3), 697–710 (2002)
7. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, Inc. (2000)

GPU Implementation of Spiking Neural Networks for Edge Detection

Zhiqiang Zhuo, Qingxiang Wu, Zhenmin Zhang,
Gongrong Zhang, and Liuping Huang

College of Photonic and Electronic Engineering, Fujian Normal University, Fuzhou, China
zhuozhiqiang1@163.com, qxwu@fjnu.edu.cn

Abstract. Spiking neural networks (SNN) are effective model inspired by neural networks in the brain. However, when networks increase in size towards the biological scale, it is time-consuming to simulate the networks using CPU programming. To solve this problem, Graphic Processing Units (GPU) provide a method to speed up the simulation. It is proposed and proved as a pertinent solution for implementation of large scale of neural networks. This paper presents a GPU implementation of SNN for edge detection. The approach is then compared with an equivalent implementation on an Intel Xeon CPU. The results show that the GPU approach provide about 37 times faster than the CPU implementation.

Keywords: Graphic processing units, spiking neural network, edge detection.

1 Introduction

The human visual system is a magical efficient image processing system. It's a hugely complex, massively parallel processors which have about 130 million neurons[1]. To simulate models of human visual system, implementation of spiking neural networks(SNN) become a significant computational technology. To date, SNNs have been simulated using different neuron models, such as the conductance based integrate and fire [2], a simple spiking neuron model [3], and Hodgkin-Huxley model [4]. These models are simulated on conventional CPU based systems and they are executed sequentially. As SNNs are highly parallel and large scale, it would take long time to simulate, despite of the increasing power of conventional CPUs. GPU implementation has provided a new solution for this problem [5,6]. Therefore, in this paper, a GPU-based implementation approach is proposed to simulate a SNN. In the approach simulation time is decreased using a parallel-mechanism. In this paper, the strategy and techniques for implementation of large scale of SNNs are addressed.

The remainder of this paper is organized as follows. In Section 2, NVIDIA GPU and the CUDA are introduced. The architecture of the SNN for edge detection is presented in Section 3. In Section 4, GPU-based implementation of the SNN model is presented. Experimental results, which obtained from implementation, are shown in Section 5. Section 6 will outline the direction of further research .

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 147–152, 2013.

2 GPUs and CUDA

Graphics Processing Unit (GPU) is a hardware unit usually used by computers to render graphic information. As we know, CPU is designed to control flow and store data cache, but differently, GPU composes of massive multi-threading cores that operate together and devotes more transistors to data processing in parallel. The GPU is especially well-suited to address problems that can be expressed as data-parallel computation in which the same program is executed on many data elements in parallel with high arithmetic intensity [7].

CUDA, which is a parallel computing engine in GPUs developed by NVIDA, supports programmers to compute in parallel like CPUs by using GPU's memory and runs thousands of lightweight threads [8]. The parallel computing engine is accessible to software developers through various kinds of industry standard programming languages, such as like C, C++ and Fortran. Programmers can use 'CUDA C'(C/C++ with NVIDA extensions in certain restrictions) to code algorithms for execution on the GPU. With the development of technology, the computational capability of GPU is dramatically improved. Multiple kernels can execute concurrently on a device, so that maximum utilization can also be achieved by using streams to enable enough kernels to execute concurrently [7]. It will help us to solve the problems in SNN simulation more easily. Tesla C2075 card with computational capability 2.0 is selected in our experiments. The card contains 448 application-acceleration cores. The Telsa processors offload parallel computations from the CPU to dramatically accelerate the floating point calculation performance. The computational time is faster than adding a second CPU. It's an unbeatable solution of SNN simulation for large scale networks in less time [8].

In conclusion, CUDA can be used to implement large scale of SNN. Since GPUs can be installed with the PC motherboard, it makes that GPU-based SNN can be applied to industry products

3 SNN Model for Edge Detection

In order to simulate the SNNs in GPU platform, a SNN model for edge detection proposed by Q.Wu et al. [9] is selected. The SNN architecture of this application will be briefly described in the rest of this section. The edge detection SNN is to use simulated using four types of receptive fields (i.e. up, down, left, right receptive fields) to detect the edges contained within the SNN input image.

The structure of the network is shown in Fig.1. Photonic receptors are represented in the first layer. Each pixel corresponds to a receptor. Every value in this layer is forwarded on to the intermediate layer 'N', via 5*5 receptive field that contains excitatory and inhibitory fields for four orientations. Excitatory and inhibitory synapses are labelled as ' X 'and ' Δ' respectively[9].

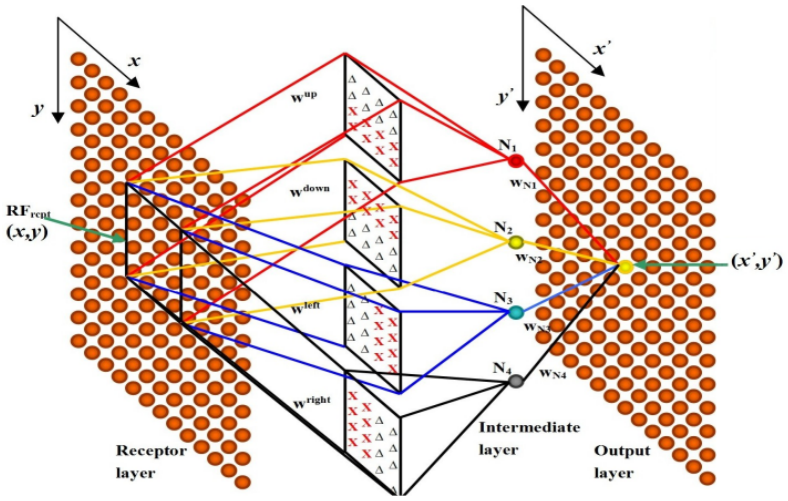


Fig. 1. SNN Model for Edge Detection(Wu et al.,2007)

In order to generate the final output layer which represents the output image, each intermediate layer N1, N2, N3, N4 is a loop. When each of the neurons in an intermediate layer fires, a firing rate can be obtained for corresponding pixels value in the final output layer. A neuron with high firing rate corresponds to edge pixel. Therefore, the firing rate can be transferred to an edge image. The algorithm is serial algorithm which is quiet time consuming because of many loops. In order to speed up the algorithm, we have implemented it using GPU programming .

4 Implementation Based on GPU

GPU solve problems using lots of data-parallel computations. Implementation of SNN models is such a problem in which the same neuron model is executed on many data elements in parallel. Each data element executes the same model and the process don't exist sophisticated flow control. GPU executes on many data elements and has high arithmetic intensity, the memory access latency can be hidden in calculations instead of big data caches. Therefore, GPU is suitable to implement SNN models.

The SNN model has been implemented in Matlab. However, it will take a very long time to detect edge when the size of image is large. Since the network model is fully parallel architecture, CUDA has potential to accelerate the algorithm. In Fig.1, it can be found that there are four parallel arrays of neurons in the intermediate layer. Each array is executed similar computation. Therefore, in this CUDA implementation, these arrays are computed at the same time by using parallelism. Independent threads are used to process the task corresponding to each pixel in the image. The parallel algorithm for detection of image using SNN is as follows.

1.Begin
2.Load input image
3.Determine image size
4.Apply RF field
5.Allocate and initialize memory on GPU for image
6.Copy input image to the GPU.
7.For each time step do
8.Execute the four parallel arrays computation kernels
9.Copy output data from the GPU
10.End

Among the steps, we can understand more detailed description of the step 8 by means of reading below- mentioned CUDA algorithm.

1.Set parameters v_{th}, E_1, g_1 , c_m, τ_{ref} , A_{ih}, A_{ex}.
2.Transfer input membrane potentials or external currents to corresponding peak conductance $q^{ex}_{x,y}$, $q^{ih}_{x,y}$
3.Allocate and initialize memory, NumGrid=width NumBlock=height, Thread Number=NumGrid*NumBlock, width*height=Pixel Number. Each thread corresponds to a pixel. Index of block and grid are corresponding to coordinates x and y.
4.Calculate N1,N2,N3,N4 neuron array $S_n(t_n)$ at the same time.

$$I = (G_{x,y} * w^{ih}_{x,y}) / A_{ih} - (G_{x,y} * w^{ex}_{x,y}) / A_{ex}$$

$$v_{x',y'}(t_n) = v_{x',y'}(t_n) + (g_l * (E_l - v_{x',y'}(t_n)) + I) / c_m$$

$v_{x'y'}(t_n)$>threshold v_{th} ,then $S_n(t_n)=S_n(t_n)+1$
5.Calculate output neuron potential $Outrate(t_n)$

$$J = (weout * S_N(t_n) / A_{ex} * (E_e - Outv(t_n))$$

$$Outv(t_n) = Outv(t_n) + (g_l * (E_l - Outv(t_n)) + J) / c_m$$

6.If $Outv(t_n)$>threshold v_{th} , then generate a spike $Outrate(t_n)= Outrate(t_n)+1$. and fall into refractory state for a period τ_{ref}
7.If $t<t_{max}$ then n=n+1:goto 2
8.End

Where NumBlock represents the number of threads in each block, NumGird represents the number of blocks in CUDA. $v_{x'y'}(t_n)$ represents membrane potentials for the four intermediate neuron arrays.

In this parallel mechanism, each image's pixel is processed by one thread. All the pixels are processed at the same time in CUDA instead of the processing in the serial algorithm by loops. The serial algorithm's asymptotic time complexity $T(n)= width* height *t_{max}$, and the parallel algorithm's asymptotic time complexity $T'(n)= t_{max}$. The speed of this part is theoretically about 100 times faster than serial algorithm. However, because there are some loops operations and "logical judgment statement" operations in the algorithm, the speed-up rate in the experiment is less than this value.

This method is a universal method. As long as the SNN contains a mass of parallel computations, it can be used, for example, SNNs for color image segmentation, discrete cosine transform and so on. For comparison, the edge detection SNN model has also been implemented using C#, and C.

5 Experimental Results

In the experiments, the following parameters for the network were used in the experiments. v_{th} = -60 mv. *vreset* = -70 mv. E_{ex}= 0 mv. E_{ih}= -75 mv. E_l= -70 mv. g_l =1.0 μs/mm^2. c_m=10 nF/mm^2. τ_{ex}=4 ms. τ_{ih}=10 ms. τ_{ref} =6 ms. The test images presents to the network. The results are shown in Fig.2.

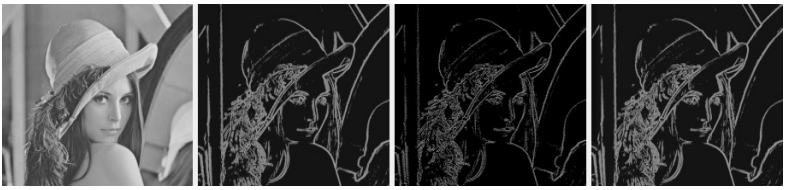


Fig. 2. Detection Edge for Multi-lineate Image

In Fig.2, the first picture is original image. The second one is the output from C#. The third one is the output from C. The last one is the output from CUDA C.

As shown in the figures, the lines show that the corresponding neurons fires with a high frequency and indicate the edges with contrast. So, using the firing rates, different contrast edges can be separated. In order to compare the running time, different size images are used to test the algorithms implemented in Matlab, C#, C and CUDA. Comparable results are shown in Table 1. These results are achieved using a PC system with Intel(R) Xeon(R) CPU E5-26200 and NVIDIA Tesla C2075. It can be seen that CUDA implement is about 37.7 times faster than C for an image with size 500*816. It is incredible faster than Matlab.

Table 1. Comparison results of run time

Platform / Image	Matlab	.NET(C#)	C	CUDA (GPU)
256*256	6817.24ms	4740ms	420ms	14ms
512*512	30279.8ms	22749ms	2415ms	61ms
482*640	33899ms	27174ms	2916ms	81ms
500*816	32042.6ms	46800ms	3996ms	106ms

For comparison, CUDA method is about 1.6 times over the Nvidia C1060 card with 4GB of DDR3 memory based implementations as reported. Therefore, there exists more parallelization in the proposed method. In Parick Dempster's method[10], only selective orientations are executed in parallel. Their final output results are not used GPU. In the proposed method, all are computed in parallel. For the picture

'lena', it takes 5109ms to execute in Parick Dempster's CUDA platform. In our platform, it's only 2207ms. However, CUDA programs still can be improved, for example, replacing the loops with CUDA programs. It is possible to speed up the algorithm further.

6 Conclusion and Future Work

This paper proposed a general implementation of SNNs using CUDA architecture in the GPU. The algorithm of SNN model is used to demonstrate the implementation techniques. The experimental results illustrates that CUDA implementation of SNNs can speed up running time over 37.7 times for images with size 500*816.

Further work is to study further reducing the running time, to apply the proposed approch to SNN architectures with more complex neuron models, and to develop a theory to bridge two architectures of SNNs and CUDAs.

Acknowledgements. The authors gratefully acknowledge the fund from the Natural Science Foundation of China (Grant No.61179011) and the Natural Science Foundation of Fujian Province (Grant No.2011J01340).

Reference

1. Xie, E.M., McGinnity, T.M., Wu, Q.X.: GPU implementation of spiking neural networks for color image segmentation. Image and Signal Processing 2011 3, 1246–1250 (2011)
2. Maguire, L.P., McGinnity, T.M., Glackin, B., Ghani, A., Belatreche, A., Harkin, J.: Challenges for large-scale implementations of spiking neural networks on FPGAs. Neurocomputing 71, 13–29 (2007)
3. Lzhikevich, E.M.: Simple Model of Spiking Neurons. IEEE Trans on Neural Networks 14, 1569–1572 (2003)
4. Gerstner, W., Kistler, W.: Spiking Neuron Models: Single Neurons, populations, Plasticity. Cambridge University Press (2002)
5. Nageswaran, J.M., Dutt, N., Krichmar, J.L.: A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphicsprocessors. Neural Networks 22, 5–6 (2009)
6. Bernhard, F., Keriven, R.: Spiking Neurons on GPUs. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3994, pp. 236–243. Springer, Heidelberg (2006)
7. CUDA 5.0 C Programming Guide, `http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html`
8. NVIDIA Tesla C2075 companion processor calculate results exponentially faster, `http://www.nvidia.co.uk/content/PDF/datasheet/NV_D_Tesla_C2075_Sept11_US_HR.pdf`
9. Wu, Q.X., McGinnity, T.M., Maguire, L., Belatreche, A., Glackin, B.: Edge Detection Based on Spiking Neural Network Model. In: Huang, D.-S., Heutte, L., Loog, M. (eds.) ICIC 2007. LNCS (LNAI), vol. 4682, pp. 26–34. Springer, Heidelberg (2007)
10. Dempster, P., McGinnity, T.M., Glackin, B., Wu, Q.X.: Performance Comparison Of a Biologically Inspired Edge Detection Algorithm On CPU, GPU And FPGA. In: International Conference on Fuzzy Computation, pp. 420–424. SCITePress, Valencia (2010)

Detecting and Recognizing LED Dot Matrix Text in Natural Scene Images

Wahyono and Kang-Hyun Jo

Intelligent Systems Lab., Graduate School of Electrical Engineering,
University of Ulsan, Ulsan, 680-749, Korea
wahyono@islab.ulsan.ac.kr, acejo@ulsan.ac.kr

Abstract. This paper addresses a method for light-emitting diode (LED) dot matrix text detection and recognition in natural scene images. Unlike general text detection and recognition, the LED text detection is quite difficult to be done due to discontinuous character. In our proposed method, first, the Canny edge detector is applied to produce an edge image. From the edge image, the interesting points representing the center of a blob are extracted. These interesting points then are merged based on their properties to generate a character component. Through feature-based template matching, the filtering and recognizing process are performed simultaneously. Experimental results show that the proposed method is reliable, effective and fast to detect and recognize the LED text in natural scene images which general text method does not cover.

Keywords: LED dot matrix, text detection and recognition, feature-based template matching.

1 Introduction

Text detection and recognition play a significant role in many applications such as environment understanding, robot navigation, information retrieval, visually handicapped assistance system, etc. On the other hand, in recent years light-emitting diodes (LED) dot-matrix text [1] (LED text for short) become widely used in destination sign on buildings, public transport vehicles, road signs or even as part of electronic devices in daily life. Consequently, text detection system which is built should be able to handle the LED text. One character of the LED text is generally displayed by a matrix of LEDs or matrix of blobs with circle or rectangular shape. Different from general text, the LED text is quite difficult to be detected because of discontinuous character. Unfortunately, though many researchers have researched text detection and recognition with significant results [2,3,11,12,13], yet rarely among those who can deal the LED text, particularly in natural images. Accordingly, in this work, we focus on this type of text.

Mainly, the text detection methods in natural images can be grouped in two categories: region-based method [4-7] and texture-based method [5]. Region-based method uses similarity characteristics of character component, such as color [4,5] ,

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 153–158, 2013.

stroke width [6], and edge [7]. By this similarity, the pixels are connected as region. Hereafter, the non-text connected components are filter out by classifier. However, this method could not effectively apply on the LED text, since a single character always consist of more than one component. Meanwhile, texture-based method utilizes texture feature of text to extract candidate of text regions by sliding windows. This method may still work on the LED text, but they could be time consuming.

In this work, we consider to overcome such problem by proposing a novel method for detecting and recognizing the LED text that would be described in section 2. Whilst, the experimental results are given in section 3. Section 4 concludes the paper and discusses about our future works.

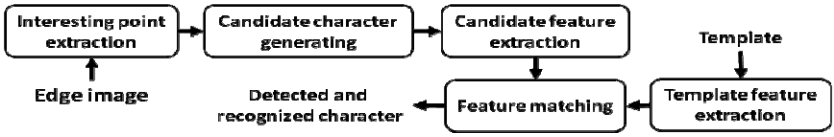


Fig. 1. The flowchart of proposed method

2 The Proposed Method

As shown in Fig. 1, the proposed method consists of four stages: *Interesting Point Extraction*: Interesting points representing the center of blobs are extracted from an edge image (Fig. 2 (c)); *Character Candidate Generating*: The interesting points are merged based on their properties to generate character candidate, called component (Fig. 2 (d)); *Feature Extraction*: After component gained, the next step is extracting some features as component characteristics. We propose a new feature based on the centroid of subcomponents; *Filtering and Recognizing*: We employ featured-based template matching to filter and recognize simultaneously every single component by comparing the component feature and template feature (Fig. 2 (e)(f)).

Fig. 2. The LED text detection and recognition process. (a) Input image, (b) edge image, (c) interesting points extraction, (d) character candidate generating, (e) feature extraction and filtering, (f) detected and recognized text marked with red circles.

2.1 Interesting Points Extraction

Instead of using circle detection [8], we prefer to propose a new technique to acquire the interesting points due to both processing time and uncertainty blob shape. We first find edges in the image using Canny edge detector [9] with threshold 175 and 320 [6].

If point $p(x,y)$ is non-edge pixel, we follow the ray $R(r,y)$, where $r=x+n;n>0$ until an edge pixel q_0 is found. Subsequently, we rotate q_0 counterclockwise by α_i degrees in order to determine point q_i with p as center of rotation, where $i=1..m-1$, $\alpha_i=2\pi i/m$, and m is the number of direction. Afterwards, the distance between p and q_0, $d_0=||p-q_0||$ as well as the distance between p and q_i, $d_i=||p-q_i||$, called radius are calculated. The number of associated points N, the mean of radius μ and the variance of distance σ^2 are defined as:

$$N=\sum_{i=0}^{m-1} f(q_i), \qquad \mu=\frac{1}{N}\sum_{i=0}^{m-1} d_i f(q_i), \qquad \sigma^2=\frac{1}{N}\sum_{i=0}^{m-1}(d_i-\mu)f(q_i) \quad (1)$$

where $f(q_i)=1$, if q_i is an edge pixel otherwise $f(q_i)=0$. As shown in Fig. 3, q_0, q_1, and q_3 are edge pixels, while q_2 is not an edge pixel. If the value of $N/m \le T_1$, $\sigma^2 \le T_2$ where T_1 and T_2 are threshold value, the point p is considered as interesting point, and the location of p, (x,y) and the value of μ are stored as properties of p.

Fig. 3. The interesting point extraction. The center image is the edge image after applying canny edge detector on the left image. In the right image, the point p is the interesting point candidate, while q_0, q_1, q_2 and q_3 are associated points of p (in this case $m=4$).

2.2 Character Candidate Generating

The output of the interesting points extraction is a set of interesting points p_i and their properties, the location (x_i,y_i) and radius μ_i. Afterwards, these points are grouped into several character components. Let denote p_1 and p_2 are the first and the second interesting point. Inspired from connected component labeling [10], point p_1 and p_2 may be grouped together as the same component if they have similar radius, $|\mu_1-\mu_2| \le T_3$, similar average color, $||AC_1-AC_2|| \le T_4$, and the distance between them is close enough, $||p_1-p_2|| \le 2*(\mu_1+\mu_2)$. The average color of p_i is defined as average colors from point p_i and its 8-neighboring pixels. For preliminary filter, the component which has interesting points less than 10 is discarded, since we assume that the character in the LED text consists of at least ten blobs. As shown in Fig. 2(d) interesting points are labeled as several components.

2.3 Feature Extraction

Normally, character component has some properties, such as border and region which pixel points form adjacent connected component. From our survey, most of character recognition methods utilize character region and border to extract feature

[2,3,11,12,13]. Unfortunately, character component in the LED text only has a set of interesting points which are not located as adjacent points, and has not character region as well as border. Though region can be formed using region growing from the interesting points, but it will take high cost in processing time. Therefore, we consider to analyze the interesting points by calculating moment and centroid.

In order to extract feature from component, firstly the centroid of component i, $Ce_i(x_i,y_i)$ is obtained. For each interesting point j in component i, p_{ij}, we calculate $d_{ij}=||p_{ij}-Ce_i||$ and α_{ij}, unsigned angle (0-2π radians) between $p_{ij}Ce_i$ and x-positive axis. Then, we classify point p_{ij} into b bins (subcomponents) based on α_{ij} voting. Thereafter, we calculate the centroid of subcomponent k ($k=1..b$), $Ce_{ik}(x_{ik},y_{ik})$. The last step, we compute $d_{ik}=||Ce_{ik}-Ce_i||$ and α_{ik}, unsigned angle (0-2π radians) between $Ce_{ik}Ce_i$ and x-positive axis, and then normalize them. With the result that, for component i, we will obtain a set of $2b$-elements vector $\{d_{i1},d_{i2},\ldots d_{ib}, \alpha_{i1}, \alpha_{i2},\ldots, \alpha_{ib}\}$.

2.4 Filtering and Recognizing

To recognize the LED text, the common text recognition method could not be performed directly, since character candidate does not contain of adjacent points as general text. Hence, in the last process of our method, the filtering as well as recognizing schema are performed by featured-based template matching. First, the character (alpha-numeric) templates are collected and are classified into 62 classes ('a-z','A-Z','0-9'). Then, we extract feature of template. Let denote S_{ij} as similarity between component i and template j that is formulated by

$$S_{ij} = w\sum_{p=1}^{b}\left(d_{ip} - d_{jp}\right)^2 + (1-w)\sum_{p=1}^{b}\left(\alpha_{ip} - \alpha_{jp}\right)^2 \quad (2)$$

where w is weighting factor. The component i may be recognized as class of template j, if the value of S_{ij} is minimal, compared with other templates. However, if the value of minimal S_{ij} is more than threshold T_5, component i will be rejected as non-text component.

3 Experimental Results

In experiment, the dataset contains *1240* character templates and *224* images with size ranging from *160x120* to *1600x1200*. The parameter values are set $b=8$, $T_1=0.6$, $T_2=1$, $T_3=1$, $T_4=1600$, and $T_5=0.65$. All parameters are empirically determined based on training templates. In order to evaluate our system, we divide evaluation category into two parts: evaluation for text detection and evaluation for text recognition. The following parameters are defined: True detected text (TDT) and false detected text (FDT) are the number of text components detected as text component and non-text component, respectively. True recognized text (TRT) and false recognize text (FRT) are denoted as the number of text components recognized as true character and false character, respectively. The last, actual text component (ATC) is

denoted as the number of text component in the ground truth. The performance measurements are defined as follows:

- Detection Recall (DR) = TDT/ATC
- Detection Precision (DP) = TDT/(TDT+FDT)
- Detection F-measure (DF) = (2xDRxDP)/(DR+DP)
- Recognition True Rate (RTR) = TRT/ATC
- Recognition False Rate (RFR) = FRT/ATC

Based on above measurements, the performance of our methods are DR=71%, DP=68%, DF=69.4%, RTR=52%, and RFR=19%. The average processing time is about 0.45s. Furthermore, Fig. 4 shows the effect of template number on our performances, while Fig. 5 shows some detection and recognition results from our proposed method. Since we have not found other work on the LED text detection and recognition in literature, we cannot compare our work with others. However, these results show that our proposed method is reliable and effective to detect and recognize the LED text.

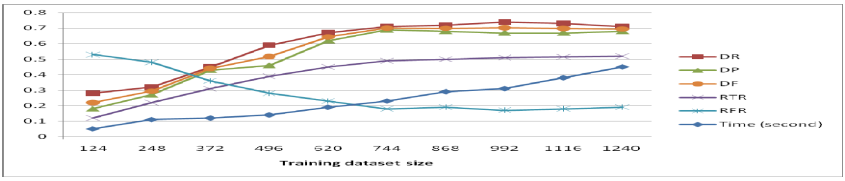


Fig. 4. The effect of template number on performance measurements

Fig. 5. Some example results of our proposed method. The results are marked with yellow circles and recognized characters. Note that the LED texts are detected in full image, but we only show cropped subimage due to space limitation. The last row shows failures of proposed method; (the first image) our method can detect the LED text, but the recognizing results are not true due to slanted text. (the second image) missing detection due to continuity of the LED text. (the remaining images) false positive due to repeating patterns.

4 Conclusions

In this paper, we propose a novel and fast method for detecting and recognizing the LED text. In our method, we utilize feature-based template matching to filter as well as recognize character component as either text or non-text component. First, we extract the interesting points on an edge image, and then they should be grouped as one component by some heuristic rules. Experimental results demonstrate the effectiveness of our method.

Nevertheless, our method still has limitations, such as handling touching character, repeating pattern and dealing with slant text. In future work, we consider to deal with these problems. Besides, we also consider to utilize another recognition technique.

Acknowledgement. This work was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the Human Resources Developtment Program for Convergence Robot Specialist support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2012-H1502-12-1002).

References

1. Huang, W.F.: Designing a display unit to drive the 8x8 dot-matrix display. In: IEEE 5th International Nanoelectronics Conferences (INEC), pp. 385–388 (2013)
2. Neumann, L., Matas, J.: A method for text localization and recognition in real-world images. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part III. LNCS, vol. 6494, pp. 770–783. Springer, Heidelberg (2011)
3. Wang, K., Babenko, B., Belongie, S.: End-to-End Scene Text Recognition. In: International Conference on Computer Vision, ICCV (2011)
4. Yi, J., Peng, Y., Xiao, J.: Color-based clustering for text detection and extraction in image. In: 15th International Conference on Multimedia (2007)
5. Liu, C., Wang, C.-H., Dai, R.-W.: Text Detection in Images Based on Color Texture Features. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005, Part I. LNCS, vol. 3644, pp. 40–48. Springer, Heidelberg (2005)
6. Epshtein, J., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Proc. CVPR (2010)
7. Zhang, J., Kasturi, R.: Text detection using edge gradient and graph spectrum. In: International Conference on Pattern Recognition (2010)
8. Cueevas, E., et al.: Fast algorithm for multiple-circle detection on images using learning automata. IET Image Processing (2012)
9. Canny, J.: A Computational Approach To Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), 679–714 (1986)
10. He, L., Chao, Y., Suzuki, K.: A run-based two-scan labeling algorithm. IEEE Transaction on Image Processing 17(5), 749–756 (2008)
11. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: Computer Vision and Pattern Recognition (CVPR), pp. 3538–3545 (2012)
12. Yi, S., Tian, Y.: Text string detection from natural scenes by structure-based partition and grouping. IEEE Transactions on Image Processing 20(9), 2594–2605 (2011)
13. Weinman, J.J., Learned-Miller, E., Hanson, A.R.: Scene text recognition using similarity and a lexicon with sparse belief propagation. IEEE Transaction on Pattern Analysis and Machine Inteligence 31, 1733–1746 (2009)

An Adaptive Controller Using Wavelet Network for Five-Bar Manipulators with Deadzone Inputs

Tien Dung Le[1] and Hee-Jun Kang[2,*]

[1] Graduate School of Electrical Engineering, University of Ulsan,
680-749, Ulsan, South Korea
dung.letien@gmail.com
[2] School of Electrical Engineering, University of Ulsan,
680-749, Ulsan, South Korea
hjkang@ulsan.ac.kr

Abstract. In this paper, an adaptive model-based control scheme is proposed for tracking control of five-bar manipulators with deadzone inputs. The proposed controller is based on the combination of nominal dynamic model of the five-bar manipulator, a wavelet network and a deadzone precompensator. The wavelet network and the precompensator are used for compensating the unknown deadzone inputs, modeling errors and uncertainties of the five-bar manipulator. The adaptation laws are derived for tuning parameters of the precompensator and wavelet network. The efficiency of the proposed control scheme is verified by comparative simulations.

Keywords: Wavelet Network, Adaptive Controller, Model-based Control, Deadzone Input, Five-bar Manipulator.

1 Introduction

In recent years, the wavelet networks have been applied in a wide range of applications in many research areas. They are a new class of networks that are inspired by both the classic neural networks and the wavelet decomposition [1, 2]. It is well-known that the wavelet networks can approximate arbitrary nonlinear functions [1, 3]. And compared with traditional neural networks, the wavelet networks can achieve the same quality of approximation with a smaller size of network. Therefore, we use the wavelet network instead of traditional neural networks for the control approach in this paper.

Deadzone inputs are often found in mechanical systems in which the actuators have some nonsmooth nonlinearities. In the past few years, control problems in mechanical systems with deadzone nonlinear inputs have attracted the attention of many researches. In [4], a fuzzy logic precompensator was proposed for deadzone compensation in nonlinear industrial motion systems. In [5], an inverse deadzone model was used as a deadzone compensation tool. This deadzone inverse technique

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 159–164, 2013.

was also used in other compensation methods in [6, 7]. In [8], an adaptive control scheme was developed by using a new description of deadzone and without constructing the deadzone inverse. This interesting idea later was used in other researches for the control problem of deadzone inputs, for example in [9, 10]. However, in most of the deadzone compensation researches above, the researchers concerned only the compensation of deadzone while other nonlinear uncertainties contained in the same plant have not been dealt with simultaneously. Therefore, in this study we also address the compensation problem of uncertainties to improve the control performance.

In this paper, we propose an adaptive control scheme for tracking control of five-bar manipulators with deadzone inputs. The proposed controller is based on the combination of nominal dynamic model of the five-bar manipulator, a wavelet network and a deadzone precompensator. The precompensator does not need the deadzone inverse model and the adaptation law is simpler than the previous methods.

The rest of the paper is organized as follows. In section 2, the dynamic model of five-bar manipulator with deadzone inputs is presented. The proposed adaptive tracking controller using wavelet network is presented in section 3. A five-bar mechanism with planned trajectories is simulated to verify the validity of the proposed controller as given in section 4. Finally, a conclusion is reached in section 5.

2 Dynamic Model

We consider the five-bar manipulators acting on a horizontal plane which were presented in [11, 12]. The dynamic model of the planar five-bar manipulator is given by [12]:

$$\hat{\boldsymbol{M}}_a\ddot{\boldsymbol{\theta}}_a + \hat{\boldsymbol{C}}_a\dot{\boldsymbol{\theta}}_a + \Delta\boldsymbol{\tau}_a = \boldsymbol{\tau}_a \tag{1}$$

where $\boldsymbol{\theta}_a = [\theta_{a1},\ \theta_{a2}]^{\mathrm{T}}$ is the active joint angle vector; $\dot{\boldsymbol{\theta}}_a = \left[\dot{\theta}_{a1}, \dot{\theta}_{a2}\right]^T$ is the active velocity vector; $\ddot{\boldsymbol{\theta}}_a = \left[\ddot{\theta}_{a1}, \ddot{\theta}_{a2}\right]^T$ is the active acceleration vector; $\hat{\boldsymbol{M}}_a \in \Re^{2\times 2}$ is the estimated inertia matrix; $\hat{\boldsymbol{C}}_a \in \Re^{2\times 2}$ is the estimated centripetal Coriolis matrix; $\Delta\boldsymbol{\tau}_a$ is the vector of modeling errors and uncetainties; and $\boldsymbol{\tau}_a \in \Re^2$ is the actuator output related to the control input $\boldsymbol{u} \in \Re^2$ through the deadzone. The detailed computations of $\hat{\boldsymbol{M}}_a$ and $\hat{\boldsymbol{C}}_a$ were presented in [12]. The vector $\Delta\boldsymbol{\tau}_a$ is presented as the following:

$$\Delta\boldsymbol{\tau}_a = \Delta\boldsymbol{M}_a\ddot{\boldsymbol{\theta}}_a + \Delta\boldsymbol{C}_a\dot{\boldsymbol{\theta}}_a + \boldsymbol{F}_a \tag{2}$$

in which $\Delta\boldsymbol{M}_a$ and $\Delta\boldsymbol{C}_a$ are the bounded modeling errors and $\boldsymbol{F}_a$ is the friction force.

The five-bar manipulator is preceded by the actuator devices which have the deadzone characteristic. Therefore, $\boldsymbol{\tau}_a$ is the actuator output not available for control and $\boldsymbol{u}$ is the actuator input which we have to design. We can describe the inputs as the following [4]:

$$\boldsymbol{\tau}_a = \boldsymbol{u} - sat_D(\boldsymbol{u}) \tag{3}$$

where

$$sat_{d_i}(u_i) = \begin{cases} d_{i+} & for\ d_{i+} \le u_i \\ 0 & for\ d_{i-} \le u_i \le d_{i+} \\ d_{i+} & for\ u_i \le -d_{i-} \end{cases} \tag{4}$$

and the diagonal matrix of deadzone width is $\boldsymbol{D} = \text{diag}(\boldsymbol{d}_1, \boldsymbol{d}_2) \in \Re^{4\times 2} = [\boldsymbol{d}_+ \boldsymbol{d}_-]^{\mathrm{T}}$.

Finally, the dynamic model of the five-bar manipulator (1) in the presence of uncertainties and deadzone inputs can be rewritten as the following:

$$\hat{\boldsymbol{M}}_a \ddot{\boldsymbol{\theta}}_a + \hat{\boldsymbol{C}}_a \dot{\boldsymbol{\theta}}_a + \Delta\boldsymbol{\tau}_a = \boldsymbol{u} - sat_D(\boldsymbol{u}) \tag{5}$$

3 Proposed Adaptive Controller

Given a desired trajectory $\theta_{da} \in \Re^2$ of the five-bar manipulator, we define the tracking error $\boldsymbol{e}$ and the filtered tracking error $\boldsymbol{r}$ as follows:

$$\boldsymbol{e} = \boldsymbol{\theta}_a - \boldsymbol{\theta}_{da} \tag{6}$$

$$\boldsymbol{r} = \dot{\boldsymbol{e}} + \boldsymbol{\Lambda e} = \dot{\boldsymbol{\theta}}_a - \dot{\boldsymbol{\theta}}_{ar} \tag{7}$$

where $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^{\mathrm{T}} > 0$ is a design parameter matrix; and $\dot{\boldsymbol{\theta}}_{ar} = \dot{\boldsymbol{\theta}}_{da} - \boldsymbol{\Lambda e}$ is defined as reference velocity vector.

The proposed adaptive tracking controller for the five-bar manipulator is presented by the following equation:

$$\boldsymbol{u} = \hat{\boldsymbol{M}}_a \ddot{\boldsymbol{\theta}}_{ar} + \hat{\boldsymbol{C}}_a \dot{\boldsymbol{\theta}}_{ar} + \hat{\boldsymbol{f}}_{WN} + \boldsymbol{u}_p - \boldsymbol{Kr} \tag{8}$$

where $\boldsymbol{u}_p \in \Re^2$ is the precompensator for the deadzone nonlinearity inputs; $\hat{\boldsymbol{f}}_{WN} \in \boldsymbol{R}^2$ is the output of a wavelet network for online learning the lump uncertainty $\Delta\tau_a$; The term $\boldsymbol{Kr}$ is used in enhancing the robustness of the control system; $\boldsymbol{K}$ is a diagonal positive definite matrix in which its elements are positive constants.

The output of wavelet network can be presented in a vector form [13]:

$$\hat{\boldsymbol{f}}_{WN} = \hat{\boldsymbol{W}}^T \hat{\boldsymbol{\Psi}}(\boldsymbol{x}, \hat{\boldsymbol{c}}, \hat{\boldsymbol{\omega}}) \tag{9}$$

where $\boldsymbol{W} = [w_{jk}]^{\mathrm{T}}$ is an $p\times 2$ matrix; $\boldsymbol{x}$ is the input vector and $\boldsymbol{\Psi} = [\Psi_1 \ \ldots \ \Psi_p]^T$. The mother wavelet functions are chosen as:

$$\Psi_j = \prod_{i=1}^{n} m_i(x_i) \exp\left\{ -\sum_{i=1}^{n} \omega_{ij}^2 (x_i - c_{ij})^2 / 2 \right\} \tag{10}$$

in which n is the number of inputs; $j = 1,..,p$ with p is the number of hidden wavelon of the wavelet network; ω_{ij} is the dilation parameter; c_{ij} is the translation parameter; $m_i(x_i)$ is chosen as:

$$m_i(x_i) = 1 - \omega_i^2 x_i^2 \tag{11}$$

The vectors of all dilation and translation parameters of wavelet basis functions are expressed as:

$$\boldsymbol{\omega} = \left[\boldsymbol{\omega}_1^T \ \boldsymbol{\omega}_2^T \ \ldots \boldsymbol{\omega}_p^T\right]^T, \ \boldsymbol{c} = \left[\boldsymbol{c}_1^T \ \boldsymbol{c}_2^T \ \ldots \boldsymbol{c}_p^T\right]^T \tag{12}$$

where

$$\boldsymbol{\omega}_j = \left[\omega_{1j} \ \omega_{2j} \ \ldots \omega_{nj}\right]^T, \quad \boldsymbol{c}_j = \left[c_{1j} \ c_{2j} \ \ldots c_{nj}\right]^T \tag{13}$$

The proposed controller (8) can be decomposed as: $\boldsymbol{u} = \boldsymbol{u}_w + \boldsymbol{u}_p$. The precompensator $\boldsymbol{u}_p$ uses the estimate $\hat{\boldsymbol{D}} = \left[\hat{\boldsymbol{d}}_+ \ \hat{\boldsymbol{d}}_-\right]^T$ of the deadzone parameter matrix $\boldsymbol{D}$ as the following [4]:

$$\boldsymbol{u}_p = \boldsymbol{\xi}\hat{\boldsymbol{d}}_+ + (\boldsymbol{I} - \boldsymbol{\xi})\hat{\boldsymbol{d}}_- \tag{14}$$

where $\boldsymbol{\xi} = \boldsymbol{I}$ if $\boldsymbol{u}_w \geq 0$ and $\boldsymbol{\xi} = \boldsymbol{0}$ if $\boldsymbol{u}_w < 0$.

The adaptation laws for online updating the parameters of the wavelet network and the deadzone precompensator are presented as the following:

$$\dot{\hat{\boldsymbol{W}}} = -\boldsymbol{\Gamma}_1 \hat{\boldsymbol{\Psi}} \boldsymbol{r}^T \tag{15}$$

$$\dot{\hat{\boldsymbol{D}}} = -\boldsymbol{\Gamma}_2 \bar{\boldsymbol{\xi}} \boldsymbol{r}^T \tag{16}$$

where $\bar{\boldsymbol{\xi}} = \left[\boldsymbol{\xi} \ \boldsymbol{I} - \boldsymbol{\xi}\right]^T$; $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_2$ are diagonal and positive constant matrices.

4 Simulation and Results

Simulation studies were conducted on Matlab-Simulink and the mechanical of the five-bar planar manipulator was built on SimMechanics toolbox following the method presented in [14].

The values of parameters in the simulation are: $l_1 =$ 0.102m, $l_2 =$ 0.18m, $l_0 =$ 0.66m, $m_1 =$ 1kg, $m_2 =$ 1.2kg, $I_{z1} =$ 0.0033kg.m^2, $I_{z2} =$ 0.0072kg.m^2, $l_{c1} =$ 0.055m, $l_{c2} =$ 0.09m. In which l_0, l_1, l_2 are the link lengths; m_1, m_2 are the masses; I_{zi1}, I_{zi2} are the inertias tensor of links of serial chain i; l_{c1}, l_{c2} are the distances from the joints to the center of mass for each link of the five-bar mechanism.

The simulations were carried out with respect to the cases when end-effector of the mechanism is driven to track a circular trajectory on XY plane. The center coordinates of the reference circular trajectory are (0.066, 0.16) and the radius is 0.05. The initial position of the end-effector **E**(x,y) of the mechanism is $\mathbf{A}_0$ (0.071,0.215). The time of simulation is 6 seconds during which the end-effector is driven to track to circular trajectory 3 times.

In the simulations, we designed three controllers to evaluate the performance of the proposed control system. *Controller 1:* The proposed adaptive controller without the wavelet network compensates for uncertainties and with the deadzone precompensator. *Controller 2*: The proposed adaptive controller with the wavelet network compensates for uncertainties and without the deadzone precompensator. *Proposed controller Eq.(8)*: The proposed controller with the wavelet network compensates for uncertainties and the deadzone precompensator.

Figure 1 shows the results of tracking control of the manipulator. The actual deadzone widths of active joint 1 and active joint 2 in the simulation are chosen as: d_{1+} = 0.8Nm, d_{1-} = 0.6Nm, d_{2+} = 0.5Nm and d_{2-} = 0.35Nm. The parameters of the filtered tracking errors are chosen as $\boldsymbol{\Lambda}$ = diag(6,6). The inputs of the wavelet network are the tracking errors and derivative of tracking errors: $\boldsymbol{x} = \left[e_{a1}\ \dot{e}_{a1}\ e_{a2}\ \dot{e}_{a2}\right]^T$. The number of neuron in hidden layer is N = 10. The learning rates are chosen as $\boldsymbol{\Gamma}_1$ = 0.5×$\boldsymbol{I}$, $\boldsymbol{\Gamma}_2$ = diag(0.0068, 0.0045, 0.0049, 0.0036). The value of design parameter matrix K is set as $\boldsymbol{K}$ = diag(10,10).

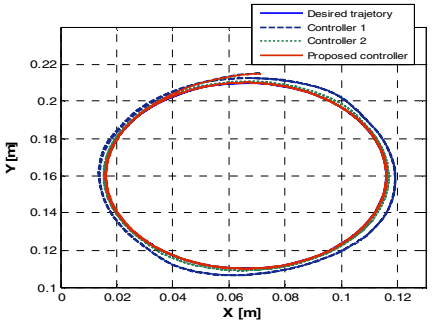


Fig. 1. The circular trajectory of tracking control

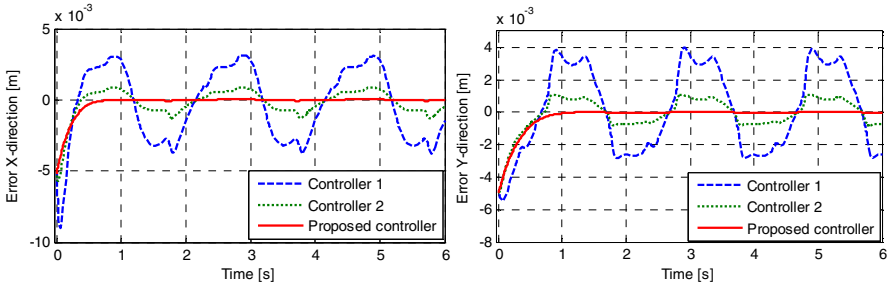


Fig. 2. Comparison of tracking errors: (a) X-direction, (b) Y-direction

The comparison of tracking errors of three cases is shown in Figure 2. As can be seen from the figure, the tracking errors caused by the controller 2 are smaller than the errors associated with the controller 1. Especially, the proposed controller brings about the smallest tracking errors compared with the controller 1 and the controller 2. It means that the proposed controller improves the tracking control performance compared to the previous results.

5 Conclusion

In this paper, an adaptive controller using wavelet network is presented for tracking control of five-bar manipulators with deadzone inputs and uncertainties. The proposed

controller is based on the combination of the dynamic model of the five-bar manipulator, a precompensator to offset the effects of deadzone inputs, a wavelet network for compensating the uncertainties, and a term for enhancing the robustness of the control system. The adaptation laws allow the wavelet network and the deadzone precompensator to be tuned online during the trajectory tracking control of five-bar manipulator. The simulation results demonstrate the effectiveness of the proposed controller in trajectory tracking control of five-bar manipulators.

Acknowledgement. The authors would like to express financial supports from Korean Ministry of Knowledge Economy under Human Resources Development Program for Convergence Robot Specialists and under Robot Industry Core Technology Project.

References

1. Alexandridis, A.K., Zapranis, A.D.: Wavelet neural networks: A practical guide. Neural Networks 42, 1–27 (2013)
2. Cao, J., Lin, Z., Huang, G.-B.: Composite function wavelet neural networks with extreme learning machine. Neurocomputing 73(7-9), 1405–1416 (2010)
3. Zhang, Q., Benveniste, A.: Wavelet networks. IEEE Transactions on Neural Networks 3(6), 889–898 (1992)
4. Lewis, F.L., et al.: Deadzone compensation in motion control systems using adaptive fuzzy logic control. IEEE Transactions on Control Systems Technology 7(6), 731–742 (1999)
5. Selmic, R.R., Lewis, F.L.: Deadzone compensation in motion control systems using neural networks. IEEE Transactions on Automatic Control 45(4), 602–613 (2000)
6. Chuxiong, H., Bin, Y., Qingfeng, W.: Performance-Oriented Adaptive Robust Control of a Class of Nonlinear Systems Preceded by Unknown Dead Zone With Comparative Experimental Results. IEEE/ASME Transactions on Mechatronics 18(1) (2013)
7. Chuxiong, H., Bin, Y., Qingfeng, W.: Adaptive Robust Precision Motion Control of Systems With Unknown Input Dead-Zones: A Case Study With Comparative Experiments. IEEE Transactions on Industrial Electronics 58(6), 2454–2464 (2011)
8. Wang, X.-S., Su, C.-Y., Hong, H.: Robust adaptive control of a class of nonlinear systems with unknown dead-zone. Automatica 40(3), 407–413 (2004)
9. Chiang, C.-C.: Adaptive Fuzzy Tracking Control for Uncertain Nonlinear Time-Delay Systems with Unknown Dead-Zone Input. Mathematical Problems in Engineering (2013)
10. Shaocheng, T., Yongming, L.: Adaptive Fuzzy Output Feedback Control of MIMO Nonlinear Systems With Unknown Dead-Zone Inputs. IEEE Transactions on Fuzzy Systems 21(1), 134–146 (2013)
11. Le, T.D., et al.: An online self-gain tuning method using neural networks for nonlinear PD computed torque controller of a 2-dof parallel manipulator. Neurocomputing (2012)
12. Le, T.D., Kang, H.-J., Suh, Y.-S.: Chattering-Free Neuro-Sliding Mode Control of 2-DOF Planar Parallel Manipulators. Internation Journal of Advanced Robotic Systems (October 22, 2013)
13. Lin, C.K.: Adaptive tracking controller design for robotic systems using Gaussian wavelet networks. IEE Proceedings Control Theory and Applications 149(4), 316–322 (2002)
14. Le Tien, D., Hee-Jun, K., Young-Shick, R.: Robot manipulator modeling in Matlab-SimMechanics with PD control and online gravity compensation. In: 2010 International Forum on Strategic Technology, IFOST (2010)

Robot Geometric Parameter Identification with Extended Kalman Filtering Algorithm

Hoai-Nhan Nguyen[1], Jian Zhou[1], Hee-Jun Kang[2,*], and Young-Shick Ro[2]

[1] Graduate School of Electrical Engineering, University of Ulsan, 680-749 Ulsan, Korea
nhan.nguyenhoai@yahoo.com, freesoulzhou@hotmail.com
[2] School of Electrical Engineering, University of Ulsan, 680-749 Ulsan, Korea
{hjkang,ysro}@ulsan.ac.kr

Abstract. This paper proposes a calibration method for enhancing position accuracy of robotic manipulators. In order to increase the robot accuracy, the method first develops a robot kinematic model and then identifies the robot geometric parameters by using an extended Kalman filtering (EFK) algorithm. The Kalman filter has advantages in identifying geometric parameters from the noisy measurements. Therefore, the obtained kinematic parameters are more precise. A simulation study of this calibration is performed for a PUMA 560 robot to prove the effectiveness of the method in increasing robot position accuracy.

Keywords: robot calibration, extended Kalman filter, parameter identification.

1 Introduction

Recently, robotic manipulators are used widely in some applications requiring high position accuracy such as off-line programming, robot-based machining, robot-aided surgery etc. In order to obtain the high accuracy, the robots should be undergone a calibration process.

There were many researches devoted to the modeling and identifying of robot error parameters [1-9, 13-14]. It is extremely difficult to model all error sources, especially non-geometric errors such as robot joint compliance, robot link deformation, gear backlash, measurement noise, random errors of joint readings etc. [6-8]. Therefore, the effects of these un-modeled errors result in inaccurate parameter identification [8-9]. Zak et. al. used a weighted least square algorithm to increase the accuracy of robot parameter identification [8]. However, the limitation of this method is that it requires information of random robot error distribution for determining the accurate weighting matrix. The extended Kalman filtering algorithm was applied widely in estimation of control system models from the noisy measurements. The application of the EKF is suitable in the robot model parameter identification due to the existence of uncertainties of measurement and modeling.

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 165–170, 2013.

In this paper, a calibration method is proposed for enhancing the robot position accuracy. The robot geometric parameters are modeled and identified by using an EKF. The Kalman filter has advantages in identifying geometric parameters from the measurements which contain uncertainties such as measurement noises and process noise (un-modeled non-geometric error sources). A simulation study is performed on a typical PUMA robot to prove the effectiveness and the correctness of the proposed method. The validation result confirms that robot has the same level of position accuracy over its workspace.

The rest of the paper is organized as follows: Section 2 presents the kinematic model of robot and derives a mathematic formulation for identifying the robot geometric errors. Section 3 formulates and applies EKF in robot parameter identification. Section 4 carries out a simulation study. Finally, a conclusion is drawn in Section 5.

2 Kinematic Model of PUMA Robot

A kinematic model of PUMA robot for calibration is developed by following [11] which used Denavit-Hartenberg (D-H) convention [10]. Frames are fixed on each robot links as in Fig. 1. The nominal D-H parameters are given in Table 1. A homogenous transformation of two consecutive link frames $\{i\text{-}1\}$ and $\{i\}$ is:

$$ {}^{i-1}_{i}\boldsymbol{T} = Rot(x_{i-1},\alpha_{i-1})Tr(x_{i-1},a_{i-1})Tr(z_i,d_i)Rot(z_i,\theta_i)\,, \tag{1} $$

where the link parameters are twist angle α_{i-1}, length a_{i-1}, offset d_i, and joint variable θ_i; $Rot(\cdot)$ and $Tr(\cdot)$ are (4×4) matrices of rotation about and translation along an axis, respectively, $i = 2\div6$.

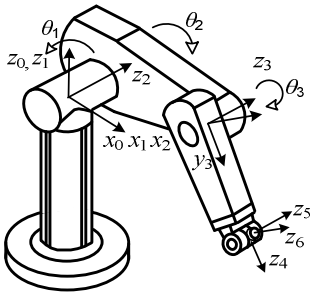


Fig. 1. Robot PUMA 560 and its link frames

A homogenous transformation matrix from the robot base frame to its end-effector frame is computed as follows:

$$ {}^{0}_{\mathrm{E}}\boldsymbol{T} = {}^{0}_{1}\boldsymbol{T}\,{}^{1}_{2}\boldsymbol{T}\,{}^{2}_{3}\boldsymbol{T}\,{}^{3}_{4}\boldsymbol{T}\,{}^{4}_{5}\boldsymbol{T}\,{}^{5}_{6}\boldsymbol{T}\,{}^{6}_{\mathrm{E}}\boldsymbol{T}\,, \tag{2} $$

where ${}^{6}_{E}\boldsymbol{T} = Tr(x_6, a_6)Tr(y_6, a_6)Tr(z_7, d_7)$ and the robot base transformation matrix is ${}^{0}_{1}\boldsymbol{T} = Rot(\mathrm{x}_0, \alpha_0)Tr(\mathrm{x}_0, \mathrm{a}_0)Rot(\mathrm{y}_0, \beta_0)Tr(\mathrm{y}_0, b_0)Rot(\mathrm{z}_1, \theta_1)Tr(\mathrm{z}_1, d_1)$. The transformation ${}^{2}_{3}\boldsymbol{T}$ of the two frames {2} and {3} (two parallel z axes) is modified as in [12] as follows: ${}^{2}_{3}\boldsymbol{T} = Rot(x_2, \alpha_2)Tr(x_2, a_2)\,Rot(y_2, \beta_2)Rot(z_3, \theta_3)$, where β_2 is the link twist parameter about the axis y_2.

A mathematical formula to identify the robot parameter errors is obtained by differentiating the transformation ${}^{0}_{E}\boldsymbol{T}$ with respect to the robot parameter errors [13]:

$$\Delta \boldsymbol{X} = \boldsymbol{J}\,\Delta \boldsymbol{p}\,, \tag{3}$$

where $\Delta \boldsymbol{X}$ is a (3×1) vector of the end-effector position errors; $\Delta \boldsymbol{p}$ is a (p×1) vector of geometric errors (p = 26, number of identifiable geometric parameters), particularly $\Delta \boldsymbol{p} = [\Delta\alpha_0 ... \Delta\alpha_5\ \Delta a_0 ... \Delta a_5\ \Delta d_1 ... \Delta d_6\ \Delta\theta_1 ... \Delta\theta_6]^{\mathrm{T}}$; $\boldsymbol{J}$ is a (3×p) Jacobian matrix; and each column of $\boldsymbol{J}$ which is corresponding to an error of $\Delta \boldsymbol{p}$ is computed as in [14].

Table 1. Nominal D-H parameters of PUMA robot (- not exits; * joint variable; × not select)

i	α_{i-1} [°]	a_{i-1} [m]	β_{i-1} [°]	b_{i-1} [m]	d_i [m]	θ_i [°] (*)
1	0	0	0	0	0	θ_1
2	-90	0	-	-	0	θ_2
3	0	0.431	0	-	0.145 (×)	θ_3
4	-90	0.02	-	-	0.433	θ_4
5	90	0	-	-	0	θ_5
6	-90	0	-	-	0	θ_6
7	0	0.3 (×)	-	0.3 (×)	0.1 (×)	-

Table 2. Assumed geometric errors of PUMA robot

i	$\Delta\alpha_{i-1}$ [°]	Δa_{i-1} [mm]	$\Delta\beta_{i-1}$ [°]	Δb_{i-1} [mm]	Δd_i [mm]	$\Delta\theta_i$ [°]
1	0.1	2	0.1	2	2	0.1
2	0.1	3	-	-	1	0.1
3	0.1	1	-0.1	-	0	0.1
4	-0.1	-1	-	-	1	-0.1
5	-0.1	-1	-	-	-1	-0.1
6	-0.1	-1	-	-	-1	-0.1

3 EKF Algorithm for Identification of Geometric Errors

A linear differential equation and a measurement equation of a constant process $\boldsymbol{Y}$ ($\boldsymbol{Y} = \Delta \boldsymbol{p}$) are expressed as follows:

$$\begin{aligned} \boldsymbol{Y}_k &= \boldsymbol{Y}_{k\text{-}1} + \boldsymbol{w}_{k-1}, \\ \boldsymbol{Z}_k &= \boldsymbol{H}_k \boldsymbol{Y}_k + \boldsymbol{V}_k, \end{aligned} \tag{4}$$

where $\boldsymbol{Y}_k$ is a n-vector of geometric errors at k^{th} measurement. $\boldsymbol{w}_{k-1}$ is vector of white noise. $\boldsymbol{Z}_k = \Delta \boldsymbol{X}_k$ is a (3×1) vector of position errors at the k^{th} measurement. $\boldsymbol{H}_k = \boldsymbol{J} \mid_{\hat{\boldsymbol{x}}_{k|k-1}}$ is a (3×n) Jacobian from (3). $\boldsymbol{V}_k$ is zero-mean measurement noise.

Equations of EKF algorithm are formulated simply as the following [2, 9]:

$$\begin{aligned} \hat{\boldsymbol{Y}}_{k\,|\,k-1} &= \hat{\boldsymbol{Y}}_{k-1\,|\,k-1}, \\ \boldsymbol{P}_{k|k-1} &= \boldsymbol{P}_{k-1|k-1} + \boldsymbol{Q}_{k-1}, \\ \boldsymbol{K}_k &= \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1}, \\ \hat{\boldsymbol{Y}}_{k|k} &= \hat{\boldsymbol{Y}}_{k|k-1} + \boldsymbol{K}_k \boldsymbol{Z}_k, \\ \boldsymbol{P}_{k|k} &= (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{P}_{k|k-1}, \end{aligned} \tag{5}$$

where $\hat{\boldsymbol{Y}}_{k\,|\,k-1}$ contains geometric errors at time k without measurement $\boldsymbol{Z}_k$; the vector $\hat{\boldsymbol{Y}}_{k\,|\,k}$ is an update of $\hat{\boldsymbol{Y}}_{k\,|\,k-1}$ with $\boldsymbol{Z}_k$; $\boldsymbol{K}_k$ is an optimal Kalman gain matrix; $\boldsymbol{Q}_{k-1} = \mathrm{E}(\boldsymbol{w}_{k-1}\boldsymbol{w}_{k-1}^T)$ and $\boldsymbol{R}_k = \mathrm{E}(\boldsymbol{V}_k \boldsymbol{V}_k^T)$ are covariance matrices.

4 Simulation Study

A simulation study is implemented to validate the effectiveness of the proposed method. A simulated PUMA robot is generated by adding the geometric errors in Table 2 with corresponding D-H parameters in Table 1. The simulated robot is imposed with geometric link errors, joint compliance errors, link deflection errors and so on. A set of 100 robot joint readings (Q_1) is generated. The robot positions are computed by its forward kinematics with the set Q_1. The robot positions should cover whole its workspace. The robot positions also are corrupted with random values (with assumption of standard distribution N $(0, \sigma)$, where σ = 0.1 mm) to simulate measurement noise.

Table 3. Residual position errors of robot (calibration)

	Mean [mm]	Max [mm]	Std. [mm]
Nominal robot model	5.6317	10.043	2.0453
After geometric calibration	0.1446	0.3840	0.0672

The robot geometric errors are identified by using the EKF algorithm on the data set Q_1. After robot geometric error compensation, its average position errors is significantly reduced to 0.1446 [mm] from 5.6317 [mm] (before calibration). More details can be seen in Table 3. The residual position errors at each measurement points are shown in Fig. 2.

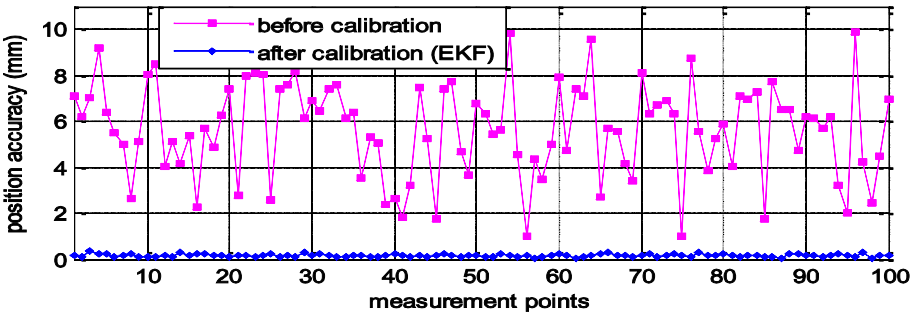


Fig. 2. Residual position errors of robot (calibration)

The robot accuracy on the set of calibration measurement points is normally higher than the other points. A set of 100 joint readings (Q_2) and an according set of end-effector position measurements are generated. The set Q_2, which completely differs from the set Q_1 and covers the overall robot workspace, is used to validate the robot position accuracy. The validation result of the proposed method is shown in Table 4. The residual position errors are shown in Fig. 3. These results show that the calibrated robot has the same level of accuracy over its workspace.

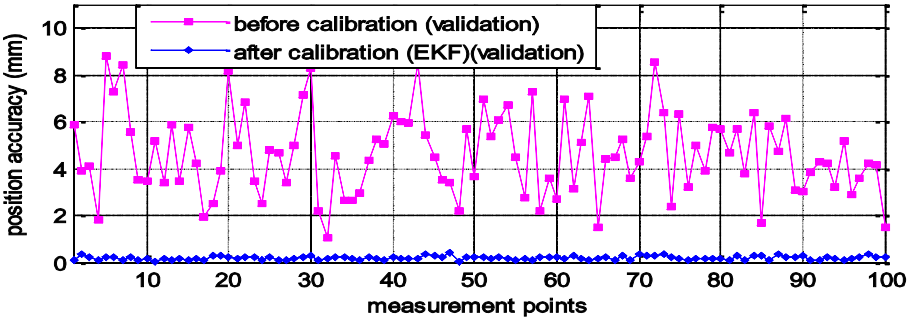


Fig. 3. Residual position errors of robot (validation)

Table 4. Residual position errors of methods (validation).

	Mean [mm]	Max [mm]	Std. [mm]
Nominal robot model	4.6280	8.7953	1.7701
After geometric calibration	0.1874	0.4238	0.0786

5 Conclusion

This paper proposed a new method for enhancing the robot position accuracy. In order to increase the robot accuracy, the method developed the robot kinematic model

and then identified robot geometric parameters by using an extended Kalman filtering algorithm. The advantage of this algorithm is that it can identify the geometric error from the noisy robot end-point measurements. The simulation results proved the effectiveness of the proposed method. In the future research, we will compensate the robot non-geometric error sources which have high non-linear relationship by using an artificial neural network.

Acknowledgments. The authors would like to express financial supports from Korean Ministry of Knowledge Economy both under Human Resources Development Program for Convergence Robot Specialists and under Robot Industry Core Technology Project.

References

1. Elatta, A.Y., et al.: An Overview of Robot Calibration. Infor. Tech. J. 3, 74–78 (2004)
2. Mooring, B.W., et al.: Fundamental of Manipulator calibration. John Wiley & Son (1991)
3. To, M., Webb, P.: An improved kinematic model for calibration of serial robots having closed-chain mechanisms. Robotica, 1-9 (2011)
4. Gong, C., Yuan, J., Ni, J.: Non-geometric error identification and compensation for robotic system by inverse calibration. Int. J. of M. Tools and Manu. 40(14), 2119–2137 (2000)
5. Judd, R.P., Knasinski, A.B.: A Technique to Calibrate Industrial Robots with Experimental Verification. IEEE Trans. on Robotics and Automation 6(1), 20–30 (1990)
6. Aoyagi, S., et al.: Improvement of Robot Accuracy by Calibrating Kinematic Model Using a Laser Tracking System, Compensation of Non-Geometric Errors Using Neural Networks and Selection of Optimal Measuring Points Using Genetic Algorithm. In: IEEE/ Int. Conf. on Intelligent Robots and Systems, pp. 5660–5665 (2010)
7. Joon, H.J., Soo, H.K., Yoon, K.K.: Calibration of geometric and non-geometric errors of an industrial robot. Robotica 19(3), 311–321 (2001)
8. Zak, G., et al.: Application of the Weighted Least Squares Parameter Estimation Method to the Robot Calibration. J. of Mechanical Design/Trans. of ASME 116, 890–893 (1994)
9. Park, I.W., et al.: Laser-Based Kinematic Calibration of Robot Manipulator Using Differential Kinematics. IEEE/ASME Trans. on Mechatronics 99, 1–9 (2011)
10. Hartenberg, R.S., Denavit, J.: A kinematic notation for lower pair mechanisms based on matrices. Trans. ASME/ J. of Applied Mechanics 77, 215–221 (1955)
11. Graig, J.J.: Introduction to Robotics: Mechanics and Control, 2nd edn. Add. Wiley (1989)
12. Hayati, S., Tso, K., Roston, G.: Robot Geometry Calibration. In: Proc. IEEE Int. Conf. on Robotics and Automation, vol. 2, pp. 947–951 (1988)
13. Veitschegger, W., Wu, C.-H.: Robot Accuracy Analysis Based on Kinematics. IEEE Journal of Robotics and Automation 2(3), 171–179 (1986)
14. Bennett, D.J., Hollerbach, J.M.: Autonomous Calibration of Single-Loop Closed Kinematic Chains Formed by Manipulators with Passive Endpoint Constraints. IEEE Transactions on Robotics and Automation 7(5), 597–606 (1991)

Improving Classification Accuracy Using Gene Ontology Information

Ying Shen and Lin Zhang*

School of Software Engineering, Tongji University, Shanghai, China
{yingshen,cslinzhang}@tongji.edu.cn

Abstract. Classification problems, e.g., gene function prediction problem, are very important in bioinformatics. Previous work mainly focuses on the improvement of classification techniques used. With the emergence of Gene Ontology (GO), extra knowledge about the gene products can be extracted from GO. Such kind of knowledge reveals the relationship of the gene products and is helpful for solving the classification problems. In this paper, we propose a new method to integrate the knowledge from GO into classifiers. The results from the experiments demonstrate the efficacy of our new method.

Keywords: Gene Ontology, Semantic Similarity, Distance Metric Learning.

1 Introduction

In the post-genomics era with the availability of large-scale gene expression data, gene function prediction becomes an emergent task. Computational approaches with novel classification techniques have been used to address this problem [3]. Despite of the success achieved by them, the improvement for the classification accuracy remains limited, because they only deal with the data obtained from the biological experiments, which contains noise and missing values. If additional information can be referred to in the prediction process, the classification accuracy should be improved. Fortunately, the Gene Ontology (GO) [9] provides us with such kind of information, which has been tentatively used for the gene function prediction [6, 14].

GO characterizes the functional properties of gene products using standardized terms. Based on GO, the semantic similarities are defined to quantitatively measure the relationships between two GO terms/gene products. Several methods have been proposed for this purpose [8, 10, 11]. Compared with the expression data, the semantic similarity information is more reliable and reflects the true relationships between the terms/gene products.

Several approaches have been proposed to make use of the semantic similarity information in the gene function prediction problems. Initially, researchers only used the semantic similarity to predict the functions for genes [7]. The problems is, because Gene Ontology is still under development, novel functions for some gene products

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 171–176, 2013.

may be masked by their known functions if the classifier only relies on the current semantic similarity information. Later, some improved methods combining both the semantic similarity and the experimental data are proposed [6, 14]. The similarities based on the expression data and the semantic similarities are weighted and together form the final combined similarities. The likelihood of a gene g having a function represented by the term t is computed using the combined similarities. Term t with the largest likelihood will be assigned to g as its potential function.

In this paper, we propose a novel method which integrates the semantic similarity information into the existing classification techniques. Specifically, in the training process, our new algorithm will learn a distance metric using the semantic similarity information. In the prediction process, classifiers can use the learned distance metric to predict functions for genes. The experimental results demonstrate that the learned distance metric can enhance the performance of the classifier.

The rest of the paper is organized as follows. Section 2 provides some background knowledge about the global distance metric learning. Section 3 introduces our new algorithm. Section 4 reports the experimental results. Finally, Section 5 concludes the paper with a summary.

2 Global Distance Metric Learning

Intuitively, the distance metric learned from the training data would be more suitable than a generic distance metric for solving a specific problem. Global supervised distance metric learning aims to solve the following problem: given a set of pairwise constraints, to find a global distance metric that best satisfies these constraints. It has been shown that the learned distance metric can significantly enhance the classifier's accuracy [4, 5].

Pairwise Constraint. can be represented by a similarity constraint set S and a dissimilarity constraint set D. Given a set of points $\{x_k \mid k = 1,\ldots, n\}$, $(x_i,\ x_j) \in S$ if x_i and x_j are in the same class; and $(x_i, x_j) \in D$ if they are in the different classes, where i, $j \in \{1, \ldots.., n\}$. Given the two sets S and D, how can we learn a distance metric that satisfies both kinds of constraints? An algorithm proposed by Xing *et al.* [12] solves this problem by minimizing the sum of distances between the samples in S:

$$\begin{aligned} &\min_{A} \sum_{(x_i, x_j)\in S} \left\| x_i - x_j \right\|_A^2 \\ &s.t. \sum_{(x_i, x_j)\in D} \left\| x_i - x_j \right\|_A \geq 1,\ A \succeq 0 \end{aligned} \tag{1}$$

A is a positive semi-definite matrix used by the Mahalanobis distance. To solve the problem formulated in Eq. (1), two solutions can be found in [12].

3 Distance Metric Learning with GO Information

In this section, we describe a novel algorithm which integrates the semantic similarity information into the existing classification technique. Specifically, in the training

process, our algorithm learns a distance metric under the supervision of a semantic similarity matrix. In the prediction process, the learned distance metric is fed into the classifier to classify the testing samples.

3.1 Distance Based on the Expression Data

Given a set of gene products $\{g_k \mid k = 1,\ldots, n\}$, the distance between a pair of gene products g_i and g_j ($i, j \in \{1, \ldots.., n\}$) is defined by the Mahalanobis distance:

$$d_{exp}(g_i, g_j) = \| g_i - g \|_A = \sqrt{(g_i - g_j)^T A (g_i - g_j)} \tag{2}$$

A symmetric distance matrix D_{exp} can be formed consequently:

$$D_{exp} = \{d_{exp}(g_i, g_j)\}_{n \times n}, i, j \in \{i,...,n\} \tag{3}$$

3.2 Semantic Similarity over Terms

Wang's method [10] is adopted here to compute the semantic similarity between terms. In [10], a GO term A is represented as $DAG_A = (A, T_A, E_A)$, where T_A is a set of terms consisting of A and all its ancestors, and E_A is a set of edges in GO that connects the terms in T_A. The contribution S of term t in T_A to term A is

$$\begin{cases} S_A(t) = 1 \text{ , if } t = A \\ S_A(t) = \max\{w * S_A(t') \mid t' \in children(t)\} \text{ , if } t \neq A \end{cases} \tag{4}$$

where w is a weight factor for the edge in E_A connecting t and t'. Given two terms A and B, the semantic similarity between them is defined as

$$sim_{Wang} = \frac{\sum_{t \in T_A \cap T_B} (S_A(t) + S_B(t))}{\sum_{t \in T_A} S_A(t) + \sum_{t \in T_B} S_B(t)} \tag{5}$$

3.3 Semantic (Dis)similarity over Gene Products

There are several approaches proposed for measuring the semantic similarity for gene products. In this paper, we propose another method to define the semantic similarity over genes. Specifically, the semantic similarity between g_1 and g_2 is defined as:

$$\begin{aligned} sim(g_1, g_2) &= \max sim(t_i, t'_j) \text{ , if } l_1 = l_2 \\ sim(g_1, g_2) &= \min sim(t_i, t'_j) \text{ , if } l_1 \neq l_2 \end{aligned} \tag{6}$$

where l_1, l_2 are the class labels for g_1 and g_2 In the training set. Using the semantic similarities computed using Eq. (6), a semantic similarity matrix S_{sem} can be formed:

$$S_{sem} = \{sim(g_i, g_j)\}_{n \times n}, i, j \in \{1,...,n\} \tag{7}$$

Because the semantic similarity value has been normalized into [0, 1], a semantic distance matrix D_{sem} can be obtained using Eq. (8).

$$D_{sem} = I_{n \times n} - S_{sem} \tag{8}$$

3.4 Algorithm

The algorithm is shown in Fig. 1. The optimization problem in step 4 is defined as

$$\min_{A} \sum_{i>j} (D_{exp}(i,j) - D_{sem}(i,j))^2 \quad s.t. A \succeq 0 \tag{9}$$

Training process

1. Calculate the semantic similarities for genes in the training set using Wang's method and form the semantic similarity matrix S_{sem} using Eqs. (6) and (7);
2. Calculate the semantic distance matrix D_{sem} using Eq. (8);
3. Calculate the distance matrix D_{exp} for the gene products in the training set using Eqs. (2) and (3);
4. Find a distance metric $\|\cdot\|_A$ that minimizes the difference between D_{exp} and D_{sem};

Prediction process

5. Classify the target genes using a knn classifier and the learned distance metric.

Fig. 1. Distance metric learning with the semantic similarity information

The convex optimization problem in Eq. (9) is solved using the gradient descent method to obtain a full matrix A. We define the cost function in Eq. (10):

$$\begin{aligned} h(A) &= \sum_{i>j} (D_{exp}(i,j) - D_{sem}(i,j))^2 \\ &= \sum_{i>j} \left[(g_i - g_j)^T A (g_i - g_j) - D_{sem}(i,j) \right]^2 \triangleq \sum_{i>j} f_{ij}^{\,2}(A) \end{aligned} \tag{10}$$

The gradient of the function $h(A)$ is

$$\nabla h = 2 \sum_{i>j} \left[f_{ij}(A) \frac{\partial f_{ij}}{\partial A} \right], \quad \frac{\partial f_{ij}}{\partial A} = (g_i - g_j)(g_i - g_j)^T \tag{11}$$

The rationale behind the algorithm is that, if the functions of the training samples have been known, the semantic similarities obtained using Eq. (6) can correctly reflect the relationships between gene products. If a global distance metric that suitably maps the expression data to D_{sem} is learned in the training process, it will alleviate the effect of noise in the expression data. Under this assumption, when using the learned distance metric in the prediction process, the classification accuracy should be improved.

4 Experiments and Results

To evaluate the performance of our algorithm, it is tested on two datasets. In the experiments, we compared the classification accuracies of the standard knn classifier and the improved knn classifier using the learned distance metric.

4.1 Data Description and Experimental Setup

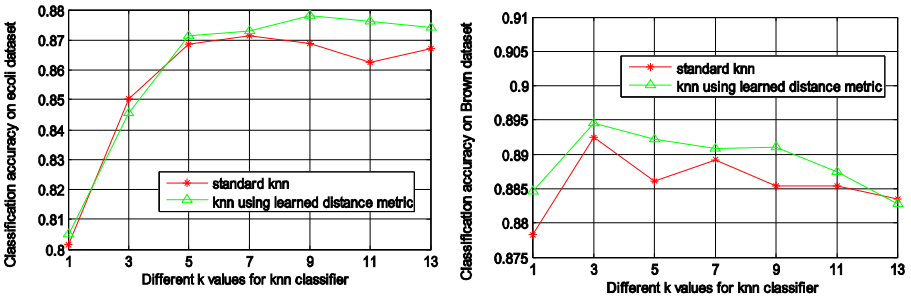
The first data set used in the experiments is the ecoli dataset from the UCI repository [1]. Annotations for gene products in the dataset were retrieved from the Uniprot database. After removing obsoleted genes in the Uniprot database, there are 309 genes left. In the experiments, only 5 classes (*cp*, *im*, *pp*, *imU*, and *om*) in which the numbers of instances are larger than 2 are used.

The second data set used is Brown's gene expression dataset (http://genome-ww.stanford.edu/clustering/Figure2.txt) [2]. The class labels can be obtained at http://compbio.soe.ucsc.edu/genex/targetMIPS.rdb. The genes are classified into 6 classes according to the MIPS function categories. Those genes that were not assigned to any of these classes and with multiple labels were eliminated. Annotations were retrieved from the SGD database. Those obsoleted genes in the SGD database were also removed. In the end, there are 224 genes left.

The semantic similarities for gene products in both datasets are computed using the *GOSemSim* package [13]. A 4-fold cross validation is performed on both datasets. We repeat the cross validation 20 times on each dataset and record the average classification accuracy for each k value.

4.2 Experimental Results

Fig. 2(a) shows the classification accuracies of the standard knn classifier and the improved knn classifier using the learned distance metric on the ecoli dataset. In this figure, the knn classifier using the learned distance metric outperforms the standard knn classifier except for the case of $k = 3$. When k is 11, the improved knn classifier outperforms the standard knn classifier by 1%. Fig. 2(b) shows the results of the experiments performed on the Brown's gene expression dataset. Again, the performance of the knn classifier using the learned distance metric is better than the standard knn classifier except for the case of $k = 13$. When k is 1, 5, and 9, the performance is improved by 0.6%.



(a) (b)

Fig. 2. Classification accuracies for the standard knn classifier and the improved knn classifier using the learned distance metric. (a) Classification accuracies on ecoli dataset; (b) Classification accuracies on Brown's gene expression dataset.

5 Conclusion

In this paper, we proposed a new method which utilizes the knowledge extracted from Gene Ontology to improve the gene function prediction accuracy by using the distance learning technique. In the training process, our method learns a global distance metric for the expression data under the supervision of the semantic similarity derived from GO. In the testing stage, the learned distance metric is used by the classifier to make decision. From the experiments, it can be seen that our method successfully improves the performance of the knn classifier, and provides a new way of integrating the GO knowledge into the classification problems in bioinformatics.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/
2. Brown, M., Grundy, W., Lin, D., et al.: Knowledge-based Analysis of Microarray Gene Expression Data by Using Support Vector Machines. PNAS 97, 262–267 (2000)
3. Guyon, I., Weston, J., Barnhill, S., et al.: Gene Selection for Cancer Classification Using Support Vector Machines. Machine Learning 46, 389–422 (2002)
4. Hinton, G., Goldberger, J., Roweis, S., et al.: Neighborhood Components Analysis. In: Proc. NIPS, pp. 513–520 (2004)
5. Weinberger, K., Blitzer, J., Saul, L.: Distance Metric Learning for Large Margin Nearest Neighbor Classification. In: Proc. NIPS (2006)
6. Pandey, G., Myers, C.L., Kuma, V.: Incorporating Functional Inter-relationships into Protein Function Prediction Algorithms. BMC Bioinformatics 10, 142–164 (2009)
7. Tao, Y., Sam, L., Li, J., et al.: Information Theory Applied to The Sparse Gene Ontology Annotation Network to Predict Novel Gene Function. Bioinformatics 23, i529-i538 (2007)
8. Resnik, P.: Semantic Similarity in Taxonomy: An Information-based Measure and Its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research 11, 95–130 (1999)
9. The Gene Ontology Consortium: Gene Ontology: Tool for the Unification of Biology. Nature Genetics 25, 25–29 (2000)
10. Wang, J., Du, Z., Payattakool, R., et al.: A New Method to Measure the Semantic Similarity of GO Terms. Bioinformatics 23, 1274–1281 (2007)
11. Wu, H., Su, Z., Mao, F., et al.: Prediction of Functional Modules Based on Comparative Genome Analysis and Gene Ontology Application. Nucleic Acids Research 33, 2822–2837 (2005)
12. Xing, E., Ng, A., Jordan, M., et al.: Distance Metric Learning, with Application to Clustering with Side-information. In: Proc. NIPS, pp. 505–512 (2002)
13. Yu, G., Li, F., Qin, Y., et al.: GOSemSim: an R Package for Measuring Semantic Similarity Among GO Terms and Gene Products. Bioinformatics 26, 976–978 (2010)
14. Yu, H., Gao, L., Tu, K., et al.: Broadly Predicting Specific Gene Functions with Expression Similarity. Gene. 352, 75–81 (2005)

A Novel Combination Feature HOG-LSS for Pedestrian Detection

Shihong Yao, Tao Wang, Weiming Shen, and Yanwen Chong*

State Key Laboratory for Information Engineering in Surveying , Mapping and Remote Sensing , Wuhan University , 129 Luoyu Road , Wuhan 430079 ,China
{yao_shi_hong,ywchong}@whu.edu.cn

Abstract. Since the ability of various kinds of feature descriptor is different in pedestrian detection and selecting them is not always fathomed, the six common features are analyzed in theory and compared in experiments. It is expected to find a new feature with the strongest description ability from their pair-wise combinations. In experiments, INRIA database and Daimler database are selected as the sample set. Adaboost is regarded as classifier and the detection performance is evaluated by detection rate, false alarm rate and detection time. The results of these three indicators further prove that description ability of HOG-LSS feature is better than others.

Keywords: pedestrian detection, feature combination, Adaboost.

1 Introduction

With the development of intelligent transportation, video surveillance and intelligent analysis, detecting pedestrian in image is a vital research topic. Pedestrian features have more variable appearance and the wide range of poses comparing with other object features. That's the reason why the pedestrian detection is a challenging task. The most critical part in pedestrian detection is how to extract effective pedestrian features on which the overall performance of detection depends.

There is much ongoing research in exploring a novel pedestrian feature for pedestrian detection. These features include Haar features[1], Scale invariant feature transform(SIFT)[2], Speeded-Up Robust Feature(SURF)[3], Histogram of Oriented Gradient(HOG)[4], Local Binary Pattern(LBP) features[5], Local Self-Similarity(LSS) features[6], Covariance features[7], Shapelet features[8] etc. The requirement of detection performance is increasing, so that a single feature no longer meets the needs of the pedestrian detection precision.

Recently, many scholars devote themselves to studying the features improvement or the features combination. Tomokia et al[9] proposed a method for extracting feature descriptors consisting of co-occurrence concept based on HOG(CoHOG). Panachit et al[10] used square-shaped detection window as the square window which can contains

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 177–181, 2013.

more variations of pedestrian. These researches show that a single feature has some limitation and deficiency, and seeking feature combination has become the trend on the study of pedestrian detection. Wang et al[11] combined the trilinear interpolated HOG with LBP as the feature set which capable of handling partial occlusion.

The detection performance of combination features is really better than the single feature's, but combining these features is not arbitrary. The paper[11] are lack of a reliable basis for them. In order to provide the theoretical basis for the selection of feature combination, we select these features (SIFT, SURF, Haar, HOG, LBP, LSS) which are most frequently used, and research their pair-wise combinations in detail. It's expected to find a combination feature with the highest detecting precision.

2 Image Feature Analysis and Feature Combination

The selected features (SIFT, SURF, Haar, HOG, LBP, LSS) all have good performance in pedestrian representation. The six features will be extracted from a typical image with size 128×64 pixels in experiments.

Table.1 shows except HOG the other five descriptors have the difference operator which is obtained by subtraction between the pixels of image blocks; SIFT, SURF and HOG descriptors are built on the basis of gradient values, and the three descriptors all need convolve with filters (here, Gaussian function and Box filter are regarded as filter). Excepting Haar descriptor, all the other five descriptors require results projection or histogram calculation. SIFT, SURF and HOG descriptors calculate histograms through the angle interval of pixels, while the LBP and LSS descriptors need to project the results into their own templates.

Table 1. Analysis of six feature descriptors

feature / operator	SIFT	SURF	Haar	HOG	LBP	LSS
Difference	√	√	√		√	√
Gradient	√	√		√		
Convolution	√	√		√		
Projection (Histogram)	√	√		√	√	√

Any two among them are combined randomly and formed new combining features. The two feature descriptors are denoted as vector A and B, and concatenated to a new feature vector. Since SURF features are deduced from SIFT features, they are not concatenated together in experiments.

According to the feature analysis from Table.1, SURF and SIFT contain all the operators and own the comprehensive image information, but they are in highly

complexity; HOG descriptors contain other three operators except difference operator. When HOG combined with Haar, LBP or LSS descriptors, the new combining features will include all the operators. That is the focus of our research in this paper.

3 Experimental

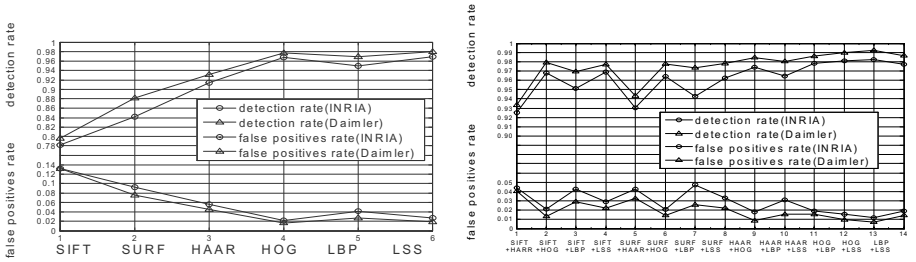
Our suggestion is demonstrated in this section. The training and testing procedures are conducted on INRIA database whose resolution is 128×64 and Daimler database whose resolution is 96×48. 2300 positive samples of frontal and other views are selected randomly in the two databases respectively, and 5000 negatives are selected randomly in INRIA database. For consistent with INRIA database, the images in Daimler are resized from 96×48 to 128×64. The adopted databases are equally split between training set and testing set randomly and we take 10-fold cross-validation in experiments. Due to the Adaboost has theoretical basis, good extensibility and the outstanding performance, it is selected as the classifier and two different criteria are adopted, the detection rate and the false positives rate, to evaluate the description ability of features.

Results are shown in Fig.1(a), HOG and LSS feature descriptors have the highest detection rate in single feature experiments, and LSS is slightly higher than the HOG. SURF and SIFT are multi-scale features and have better performance in object matching, but only selecting features of keypoints no longer meet the pedestrian detection's requirements. The constructions of Haar feature are simpler than the other features and its descriptive ability is inferior to HOG, LBP and LSS features. On the Daimler and INRIA databases, the curves' tendency of the detection rate is roughly identical and the curve of INRIA is under the Daimler's because the pedestrians in the INRIA database are much more complicated and varied. The false positives rate of the HOG features is the smallest among these six kinds of feature and the false positives rate of the LSS features is very close to it.

In the light of comprehensive consideration, we suggest to combine the HOG and LSS feature descriptors as a new descriptor. For certificating our suggestion and evaluating the performance of the combination features, any two among these six kinds of features are combined randomly. Fig.1(b) shows the detection rate comparison among the features on INRIA and Daimler database. The curves' tendency of the detection rate is roughly identical and the curve of Daimler database is above the INRIA's. Detection performances of combining features are superior to any single feature. The detection rate of HOG-LSS feature is the highest among all the feature combinations. The false positives rate comparison shown in Fig.1(b) further verifies the HOG-LSS feature is the optimal combination.

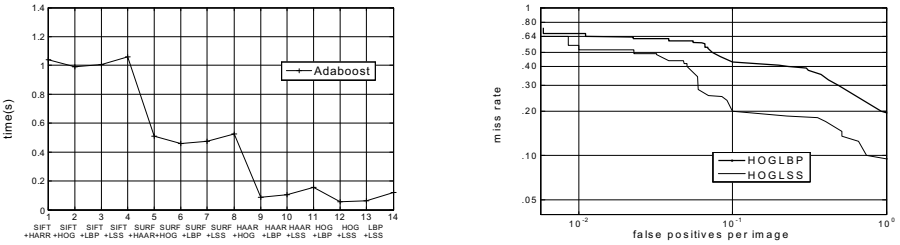
In experiments, the detection time of these combination features are considered. Fig.2(a) shows that the detection time of HOG-LBP is 58ms and the HOG-LSS is 63ms respectively. They are considered to be the same. We do not optimize speed in experiment procedure purposely. All the experiments are running on Intel(R), Core(TM), dual-core i3 processor, 3.07GHz, RAM 2.99G, Matlab2009a. If using the GPU acceleration, the speed of the pedestrian detection can be further improved.

As shown in Fig.2(b), our HOG-LSS feature is superior to HOG-LBP which is proposed in paper[12] (miss rate=1- detection rate). According to the results of these combination features and overall consideration of the detection rate, the false alarm rate and the detection time, the HOG-LSS feature is the best feature descriptor among these combining features. In theory HOG-LSS feature has a good performance, because it contains these four operators: gradient, difference, convolution, projection, which are complementary with each other.



(a) (b)

Fig. 1. (a)The detection rate comparison and the false positives rate comparison among the six features on INRIA and Daimler database and Adaboost is used as the classifier,(b) The detection rate comparison and the false positives rate comparison among the combination features on INRIA and Daimler database and Adaboost is used as the classifier.



(a) (b)

Fig. 2. (a)The detection time on the Adaboost classifier, (b) The comparison between HOG-LSS and HOG-LBP[11]

4 Conclusion

We study these common features: SIFT, SURF, Haar, HOG, LBP, LSS, which have their own application characteristics in object description, and expected to find a new combination feature with the strongest description ability. First, the six kinds of feature are analyzed in theory and experiments, and the HOG-LSS is considered to be a new pedestrian detection feature containing all the image descriptive operators. For further

verifying these, any two among the six features are combined on the INRIA and Daimler databases respectively and training them by the Adaboost classifiers. The results of experiments in three performance indicators show the HOG-LSS feature is superior to other feature combination and expected to replace the HOG-LBP feature which commonly used in pedestrian detection research currently.

Acknowledgment. This paper was supported by Natural Science Foundation of Hubei Province of China(2012FFB04204), National Natural Science Foundation of China(41271398).

References

1. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: IEEE Proceedings of International Conference on Computer Vision, pp. 734–741 (2003)
2. Lowe, D.G.: Distinctive image features from scale-invariant key points. In: IEEE International Conference on Intelligent Vehicle Symposium, pp. 19–24 (2004)
3. Bay, H., Ess, A.: Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding 110(3), 346–359
4. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 886–893 (2005)
5. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29(1), 51–59 (1996)
6. Shechtman, E., Irani, M.: Matching Local Self-Similarities across Images and Videos. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 1–8 (2007)
7. Wu, B., Nevatia, R.: Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. In: IEEE Conference on Computer Vision, pp. 90–97 (2005)
8. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 1–8 (2007)
9. Watanabe, T., Ito, S., Yokoi, K.: Co-occurrence Histograms of Oriented Gradients for Pedestrian Detection. In: Wada, T., Huang, F., Lin, S. (eds.) PSIVT 2009. LNCS, vol. 5414, pp. 37–47. Springer, Heidelberg (2009)
10. Kittipanya-ngam, P., Lung, E.H.: HOG-Based Descriptors on Rotation Invariant Human Detection. In: Koch, R., Huang, F. (eds.) ACCV 2010 Workshops, Part I. LNCS, vol. 6468, pp. 143–152. Springer, Heidelberg (2011)
11. Wang, X.: An HOG-LBP Human Detector with Partial Occlusion Handling. In: IEEE 12th International Conference on Computer Vision, pp. 32–39 (2009)

Segmentation of Slap Fingerprint Images

Kamlesh Tiwari, Joyeeta Mandal, and Phalguni Gupta

Department of Computer Science and Engineering,
Indian Institute of Technology Kanpur, Kanpur 208016, India
{ktiwari,joyeetam,pg}@cse.iitk.ac.in

Abstract. This paper proposes a novel technique to segment fingerprints from 4-slap image. Black pixels of down scaled and binarized slap images are subjected to experience a mutual superposition attractive force which is inversely proportional to the square of distance. Pixels having high force are grouped based on their force angles, and adjacent connected groups are merged to form components. These components are used to identify hand and to label fingers. The technique have been tested on IITK-Rural and IITK-Student databases and has achieved high segmentation accuracy of 96.80% and 99.2% respectively.

Keywords: Biometrics, Fingerprint, Minutiae, Slap Scanner, Segmentation, ROC curve.

1 Introduction

Fingerprints are widely accepted biometric for human identification. Ridge and valley structure available on human finger skin forms a specific patterns called fingerprint. They are unique for every individual and they contains 150 types of features like ridge ending, ridge bifurcations, core points, loops etc which are called minutiae [1]. Out of which ridge ending and bifurcation are the most prominently used. For matching two fingerprints, similarity between these minutiae defined by their location and orientation information, are compared [2,3]. Accuracy of fingerprint based systems increase when more than one finger is used for matching. Some specific scanners called four slap scanner have large scanning area that captures all four fingers simultaneously and produce an image called four slap fingerprint image. However one needs to design an efficient technique for the segmentation of fingerprints from the slap image. All contemporary matching algorithms work on the features extracted from a single fingerprint at a time. So, determining the exact location of each individual finger in a slap image along with its correct mapping to one of index, middle, ring and little fingers is necessary. Defects that poses challenges in segmentation of the slap fingerprint images are missing fingers, cropped fingers, compressed fingers, unusual orientation, halo effect and latent fingerprints.

Slap segmentation has attracted considerable amount of research interest in the last decade. The system proposed in [4] makes uses histograms along heuristically selected lines to decide finger orientation and considering local minima/maxima of the pixel histogram determines corresponding finger center and inter-finger boundaries.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 182–187, 2013.

In [5], disjoint components and their orientation are calculated based on geometrical properties of human hand. This algorithm performs well for lower hand's rotation. Technique proposed in [6] assumes elliptical shape of fingerprints and uses refined version of meanshift algorithm to identify the maximally dense, centrally located points within each finger component in slap image and then applies ellipse fitting. Another work using Fourier spectrum and knuckle line information is devised in [7]. This technique yields better performance than [6], as stated in [8] which introduces a modified version of [7] involving principal direction for each fingerprint blob such that its rotational inertia is minimized. In [9] connected components are identified by row major traversal and grouping 4-connected pixels. It detects hand type using size and position of the finger clusters. Work proposed in [10] uses pixel entropy to determine the different components of fingers. It uses PCA to estimates the orientation of component. The method is rotation invariant and does not assume elliptical shape for finger components. Limitation includes its assumption the existence of four finger tips in slap image.

Paper proposes a novel segmentation technique to extract individual fingerprints from a four slap fingerprint image. The technique is robust to noise and performs better than other existing algorithms as it can segment images having very close fingers. The paper is divided into four sections. Section 2 discusses the proposed segmentation technique. Experimental results for two databases have been analyzed in Section 3. Conclusions are presented in the last section.

2 Proposed Technique

This section describes the proposed technique to segment correctly four slap fingerprint images. Functionally, the whole technique can be divided into four major modules which are preprocessing, finger component determination, hand detection and classification or labeling of the fingers. Preprocessing involves down-scaling and binarization of the image. The component determination includes clustering the finger components. Hand detection phase identifies whether the given slap image belongs to left or right hand. This is followed by pruning of invalid clusters and assigning finger labels viz index, middle, ring or little finger to the components. The block diagram of the proposed system is shown in Fig. 1.

2.1 Preprocessing

Input slap image is down-scaled using bi-cubic interpolation to one tenth of its length and width to reduce the computation time. This down-scaled image is binarized by using zeroth and first-order cumulative moments of the intensity-level histogram [11].

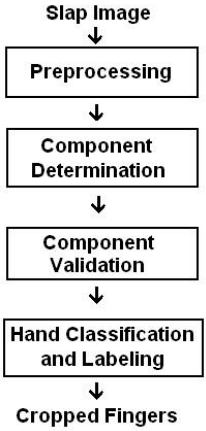


Fig. 1. Block Diagram of Proposed System

2.2 Components Determination

The first step in the component detection is the modeling of every foreground pixel of binarized image as a unit mass particles lying in free space and exerting superimposable attractive vector force on each other. The magnitude of the attractive force is inversely proportional to the square of the Euclidian distance r between them and the direction is along the line joining the particles $F = 1/r^2$. Resultant force experienced on a particle is the vector sum of all forces exerted by other particles. Point surrounded by many points experiences negligible force as compared to the boundary points of the blob. Therefore, a threshold is selected heuristically to identify near boundary points experiencing high force. Fig. 2(a) shows a slap fingerprint image with marked boundary points.

Boundary points are partitioned into eight classes on the basis of direction of their force. Class i, $1 \leq i \leq 8$ contains the points having force directions lying between $(i-1)*45^o$ and $i*45^o$ as shown in Fig 2(b)-2(i). It can be observed that every class contains well separated group of points. Points in the same group lie closer as compared with points of other group. Number of groups formed in a class vary from image to image. Points within a group form an arc sweeping an angle of 45^o on the boundary of a finger component. Points in eight nearby arcs fully enclose a finger component as shown in Fig. 2(j). It can be observed that all these eight groups of points in arcs belong to different classes. An arc in class i can have its neighboring arc either in class $i+1 \pmod 8$ or $i-1 \pmod 8$ as shown in Fig. 2(k). Thus, boundaries of all the different finger components present in a slap image are obtained.

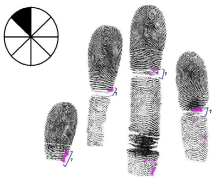
2.3 Component Validation

Detected components may contain some small spurious component which are formed due to environmental conditions and noise. In addition to this some small components

(a) Marked Boundary Points.

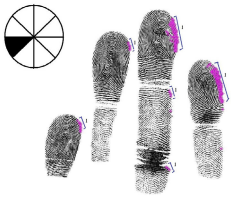
(b) Class 1

(c) Class 2

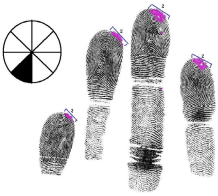


(d) Class 3

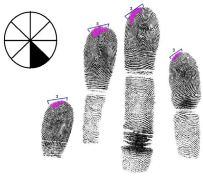
(e) Class 4



(f) Class 5



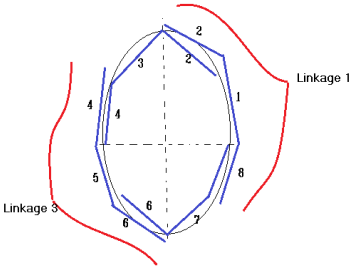
(g) Class 6



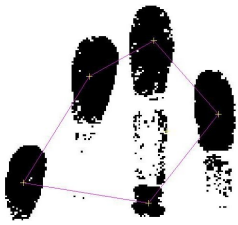
(h) Class 7

(i) Class 8

(j) Point arc on component



(k) Local linkage within a connected component.



(l) Convex hull to determine fingertips.

Fig. 2. Steps Involved in Proposed Segmentation Technique

are formed due to bad quality of fingerprint. These erroneous components are removed in this validation stage. Towards this if a component has area less than the one-fifth of the mean area then this component is removed for further consideration.

2.4 Hand Classification and Labeling

Convex hull of midpoints of remaining components is computed. The top four components are corresponding to the fingertips; as shown in Fig. 2(l). Classification of hand as left or right hand is done based upon the geometric properties of the hand. Consider the component having largest Y coordinate value if it has two components in right and one in left that the hand is a right hand, otherwise hand is left. Length of middle and little fingers is also used as the criterion for hand type identification as little finger area is smaller than middle finger one. After hand identification four fingertip components are labeled as index, middle, ring and little finger using hand geometry.

3 Experimental Results

The proposed technique has been tested against two four slap fingerprint image databases viz IITK-Student and IITK-Rural. Slap fingerprint images are obtained using optical scanner of uniform single prism scan area of *81x76* mm^2 at resolution of 500 ppi. IITK-Student database consists of fingerprint data collected from students. The database has been collected in two phases with a gap of two months. 1007 participants have provided three images of each hand at every phase. As a result database contains 12084 images. IITK-Rural database is collected from rural people, actively involved in laborious jobs like farming. Data has been acquired in non-controlled environment and in two phases with the gap of two months. 991 participants provided three images of both hand in each phase. Therefore, IITK-Rural database contains total 11892 slap images.

Results are manually examined to validate the segmentation. Accurate detection of hand type is an important preliminary step towards correct finger labeling. Among all 23976 slap fingerprint images of the two databases, 112 images could not be correctly identified as left or right hand. 11951 images are correctly identified as left hand while 11913 images were correctly identified as right hand. When the length of middle and little fingers is used as the criterion for hand type identification, hand images with missing fingers could not be classified. In 34 images having an orientation angle of more than 15^o or the having fingers missing it has been observed that the wrong component is picked as a fingertip. And images having the index and middle finger tip at the same level also failed in this step. Thus the technique has identified the correct hand with an accuracy of 99.53%.

The proposed technique has accurately segmented 11512 slap images collected in IITK-Rural database of 11892 four slap fingerprint images. It has achieved an accuracy of 96.80%. For IITK-Student database of 12084 images it has correctly segmented 11987 images and achieved an accuracy of 99.2%.

4 Conclusions

In this paper an efficient technique for slap fingerprint segmentation has been proposed. The technique has used pseudo superimposed mutual attractive force between foreground pixels of the slap image to progress. Near boundary pixels of a finger are identified based on force magnitude and then depending on the direction of this force they are divided into eight classes. Pixel location is utilized to determine group among the classes and across class neighborhood of group is used to segregate finger components. Positions of these components are used to estimate hand type and fingers labeling based on hand geometric properties. The proposed technique has been tested on IITK-Student and IITK-Rural databases having 12084 and 11892 slap images respectively. It has been observed that the proposed technique has achieved segmentation accuracy of 96.80% on IITK-Rural database and 99.2% on IITK-Student database.

Acknowledgements. Authors like to acknowledge the support provided by the Department of Information Technology, Government of India to carry out this research work.

References

1. Moenssens, A.: Fingerprint Techniques. Chilton Book Company (1971)
2. Singh, N., Tiwari, K., Nigam, A., Gupta, P.: Fusion of 4-slap fingerprint images with their qualities for human recognition. In: World Congress on Information and Communication Technologies (WICT), pp. 925–930 (2012)
3. Tiwari, K., Arya, D.K., Gupta, P.: Designing palmprint based recognition system using local structure tensor and force field transformation for human identification. Neurocomputing 6839, 602–607 (2012)
4. Ulery, B., Hickline, A., Watson, C., Indovina, M., Kwong, K.: Slap fingerprint segmentation evaluation. slapseg04 analysis report (2005)
5. Lo, P., Sankar, P.: Slap print segmentation system and method, US Patent 7, 072, 496 (2006)
6. Hodl, R., Ram, S., Bischof, H., Birchbauer, J.: Slap fingerprint segmentation. In: Computer Vision Winter Workshop (2009)
7. Yong-liang, Z., Yan-miao, L., Hong-tao, W., Ya-ping, H., Gang, X., Fei, G.: Principal axis and crease detection for slap fingerprint segmentation. In: 17th IEEE International Conference on Information Processing (ICIP), pp. 3081–3084. IEEE (2010)
8. Zhang, Y.L., Xiao, G., Li, Y.M., Wu, H.T., Huang, Y.P.: Slap fingerprint segmentation for live-scan devices and ten-print cards. In: International Conference on Pattern Recognition (ICPR 2010), pp. 1180–1183 (2010)
9. Singh, N., Nigam, A., Gupta, P., Gupta, P.: Four slap fingerprint segmentation. In: Huang, D.-S., Ma, J., Jo, K.-H., Gromiha, M.M. (eds.) ICIC 2012. LNCS, vol. 7390, pp. 664–671. Springer, Heidelberg (2012)
10. Gupta, P., Gupta, P.: Slap fingerprint segmentation. In: Biometrics: Theory, Applications and Systems (BTAS), pp. 189–194 (2012)
11. Otsu, N.: A threshold selection method from gray-level histograms. Automatica 11(285-296), 23–27 (1975)

Multimodal Personal Authentication System Fusing Palmprint and Knuckleprint

Aditya Nigam and Phalguni Gupta

Department of Computer Science and Engineering,
Indian Institute of Technology Kanpur
Kanpur 208016, India
{naditya,pg}@cse.iitk.ac.in

Abstract. This paper proposes a multimodal personal authentication system which makes use of palm and knuckleprint traits. Biometric images are enhanced and robust corner features are tracked using Lukas and Kanade tracking. Matching score between feature vectors of two images is obtained through a similarity measure which makes use of geometrical and statistical characteristics. The proposed system is tested on chimeric multimodal databases created by fusing two publicly available palmprint databases CASIA and PolyU along with PolyU knuckleprint database. Experimental results reveal correct recognition rate of 100% with EER less than 0.1%.

Keywords: Multimodal, Knuckleprint, Palmprint, Corner Features, Local Binary Pattern, Sobel Derivative, Lucas and Kanade Tracking.

1 Introduction

Behavioral as well as physiological biometrics based characteristics (such as face [14], fingerprint, iris [7,15], palmprint [6], knuckleprint [13,5], gait, voice,vein patterns etc.) are used to develop robust, accurate and highly efficient personal authentication systems. In past few years, society have noticed great attention in biometric recognition systems which are based on hand (e.g. palm print, fingerprint and finger knuckleprint) because of their low cost acquisition sensors, high performance, higher user acceptance and lesser user cooperation. There exist several unimodal palmprint [6] as well as knuckleprint [13] based systems. Most of these systems are sensitive to noisy sensor data and non-universality of both palm and knuckleprint images. Hence they can only achieve low or middle range of security. A multimodal system makes use of more than one trait to enhance system's performance, especially when the size of gallery database is large because false acceptance rate grows geometrically with the increase in database size [4].

Not much research work over palm and knuckleprint fusion is reported. In [11], features which are automatically detected by tracking are encoded using efficient directional coding and ridgelet transforms are used for matching. In [10], 1D gabor filters are used for extracting features from knuckle as well as palmprint. In [16], sharp edge like features for knuckleprint images are denoised using wavelet whereas

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 188–193, 2013.

for palmprint corner features with their local descriptors are used. Matching is done by cosine similarity function and hierarchical hand metric. In [12] radon and haar transforms are used for feature extraction of palm and knuckleprint images respectively and nonlinear fisher transformation is applied for dimensionality reduction. Finally matching is done using parzen window classification. In [9], score level fusion is preformed on palm and knuckleprint images by using phase only correlation (POC) function.

In this paper the region of interests (ROI) from palm and knuckleprint images are preprocessed using a novel sign of local gradient (SLG) method. Features are extracted by calculating Hessian matrix's eigen values at every pixel and matching is performed using the proposed HCF similarity measure. Finally scores obtained for both traits are fused to get multimodal fusion score. This paper is organized as follows: Section 2 describes the proposed recognition system. Section 3 presents the experimental results followed by the last section that concludes the presented work.

2 Proposed System

The proposed authentication system works in following phases: ROI extraction, image preprocessing, feature extraction, feature level matching and finally score level fusion. One can use the method suggested in [6,18] to extract the region of interest (ROI) from palm and knuckleprint images. These images are preprocessed by sign of local gradient (SLG) method and corner features [17] are considered for matching using the proposed HCF similarity measure. Finally, the scores obtained from both traits are fused to obtain multimodal fusion score by giving equal weights. Tasks performed in each phase are explained as follows.

[A] Preprocessing using Sign of Local Gradient (SLG): The ROI of extracted palmprints are normalized to 100×100 where as knuckleprints are normalized to 100×50 size taking their aspect ratio into consideration. The edgemaps are considered to be worked on so as to achieve robustness against illumination. Every ROI is transformed into its *gradientcode* (as shown in Fig. 1) that is assumed to be robust against illumination and small amount local non-rigid distortion. The *gradientcode* for any image P can be obtained by evaluating *sign_code* for every image pixel $P_{j,k}$ and replacing its gray value by it. The sign_code for any pixel is a 8 bit binary number whose i^{th} bit is defined as

$$sign\_code_i = \begin{Bmatrix} 1 & if\ (Neigh[i] > 0) \\ 0 & otherwise \end{Bmatrix} \quad (1)$$

where *Neigh[i]*, *i*=1,2,...8 are the gradient of 8 neighboring pixels centered at pixel $P_{j,k}$ obtained by applying scharr edge detection kernel. In *gradientcode* (as shown in Fig. 1), every pixel is represented by its *sign_code* which is just an encoding of edge pattern around its 8-neighborhood. The basic assumption is that the pattern of edges within 8-neighborhood of any pixel does not change abruptly; hence in *sign_code* of

any pixel, only the sign of the derivative in its neighborhood is considered. This property ensures robustness of the proposed system in illumination varying environments.

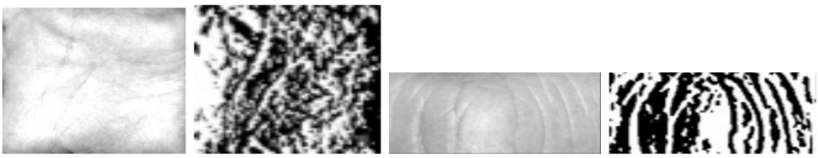


Fig. 1. Original and Transformed (*gradientcode*)

[B] Feature Extraction Using KLT Corner Detector [17]: Corner features that can be tracked accurately even in varying illumination condition as they have high derivatives in two orthogonal directions. Hence, they can provide enough robust information for tracking and can be considered as features. The KLT [17] operator which does eigen analyses of 2×2 Hessian matrix (H) at every pixel with respect to a local neighborhood of w×w is used to detect corners. The Hessian matrix **H**, can have at-most two eigen values λ_1 and λ_2 such that $\lambda_1 \geq \lambda_2$ with e_1 and e_2 as their corresponding eigenvectors. Like [17], all pixels having $\lambda_2 \geq \mathrm{T}$ are considered as corner feature points.

[C] Feature Matching Using KL Tracking [8]: Let I_a and I_b be two knuckles or palmprints that are to be matched and A, B be their corresponding *gradientcode*. Corner features in A are computed and tracked in B using KL tracking [8] which is based on three assumptions, namely *brightness consistency, temporal persistence* and *spatial coherency*. The performance of tracking depends on how well these three assumptions are satisfied. It is observed that the above mentioned assumptions are well satisfied for genuine matching while for imposters they fail. Hence tracking performance is assumed to be good when the matching is genuine. A novel similarity measure, HCF (Highly Correlated Features) estimates the performance of KL tracking in terms of some geometric and statistical quantities calculated for each tracked feature. Any feature in A successfully tracked in that of B is assumed to be highly correlated if the geometrical and statistical constraints are satisfied. These constraints are defined below.

[C.1] Geometrical and Statistical Constraints: The Euclidean distance between any feature and its estimated tracked location should be less than or equal to an empirically selected threshold TH_d. The tracking error (T_{error}) defined as pixel-wise sum of absolute difference between a patch centered at corner and that of its estimated tracked location patch should be less than or equal to an empirically selected threshold TH_e. The phase only correlation (POC) [9] between a local patch centered at any feature and that of its estimated tracked location patch should be more than or equal to an empirically selected threshold TH_p.

[C.2] Matching Algorithm: In Algorithm 1, *gradientcode* A is compared with that of B and number of highly correlated features *hcf*(A,B) of A with that of B is returned. The value of *hcf*(A,B) may not be same as *hcf*(B,A) as it is not a symmetric measure. Hence the proposed similarity score *HCF*(A,B) is calculated by taking the average of *hcf*(A,B) and *hcf*(B,A) in order to make similarity measure symmetric.

Algorithm 1 $hcf(A, B)$

Require:
The two $gradientcode$ A, B of knuckle or palmprint ROI.
Ensure: The highly correlated features $hcf(A, B)$ in $gradientcode$ A with that of B.
1. $tnc = 0$;[Counter for total number of corners in A]
2. $hcf = 0$;[Counter for number of highly correlated features in A with that of B]
for all corners $A(i, j) \in A$ **do**
Let $B(k, l)$ be its tracked location in B;[Estimated by KL Tracking]
if $((||A(i, j), B(k, l)|| \leq TH_d) \&\& (T_{error} \leq TH_e) \&\& (POC \geq TH_p))$ **then**
$hcf = hcf + 1$;[T_{error}, POC as defined in Section 2 [C.1]]
end if
$tnc = tnc + 1$;
end for
return $\frac{hcf}{tnc}$

3 Experimental Results

The performance of the system is measured using correct recognition rate (CRR) in case of identification and equal error rate (EER) for verification.

Databases: The proposed system is tested on two publicly available benchmark palmprint databases CASIA [2] and PolyU [3] and the largest publicly available PolyU [1] knuckleprint database. There are very few subjects in both CASIA and PolyU palmprint databases with incomplete or missing data such subjects are discarded for this experiment. The palmprint and knuckleprint images obtained from same subject are assumed to be totally uncorrelated with each other hence one can create chimeric multimodal databases by fusing them. The two multimodal datasets A1 and A2 are generated by fusing two modalities (i.e. palmprint and knuckleprint) of all subjects in Casia and PolyU palmprint database with that of PolyU knuckleprint database respectively. The detailed specification of unimodal as well as multimodal datasets are given in Table 1.

The HCF measure computes the similarity between two *gradientcode* and it is primarily parameterized by three parameters TH_e, TH_p and TH_d as explained in Section 2 [C.1], that are optimized over small validation sets (constituting only first 50 subjects) of each database so as to maximize the performance in terms of CRR and EER. The results obtained for unimodal and multimodal systems are presented in Table 1. It has been observed from Table 1 that the proposed multimodal system performs very well and achieve CRR as high as 100.00% with an EER less than 0.1% for both multimodal databases. For both multimodal databases (A1 and A2), Receiver Operating Characteristics (ROC) curves are also shown in Fig. 2 along with their unimodal versions.

Table 1. Experimental Results of the proposed Unimodal and Multimodal system

	Total Sub	Total Pos	Total Images	Train Poses	Test Poses	CRR (%)	EER (%)
Dataset	**UNIMODAL**						
PALM (CASIA)	566	8	4528	First 4	Last 4	99.96	0.29
PALM (POLYU)	386	20	7720	First 10	Last 10	99.95	1.49
KNUCKLE (POLYU)	660	12	7920	First 6	Last 6	99.41	3.06
Dataset	**Traits Fused**	**MULTIMODAL**					
A1	PALM(CASIA), KNUCKLE(POLYU)	566	8	566*8*2 = 9056	First 4	Last 4	100.0	0.02
A2	PALM(POLYU), KNUCKLE(POLYU)	386	12	386*12*2 = 9264	First 6	Last 6	100.0	0.12

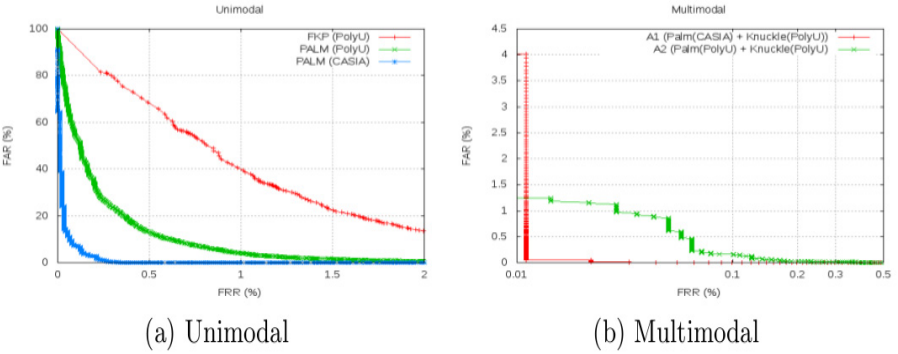


Fig. 2. ROC graphs for Unimodal and Multimodal Databases (Db name Table 1)

4 Conclusion

This paper has presented a multimodal biometric system. It uses Highly Correlated Features (HCF) similarity measure to compare structural features in palm and knuckleprints. The corner features obtained from palm and knuckleprint ROI's are tracked using KL tracking algorithm and HCF measure is computed to obtain the matching scores. It has been tested on multimodal databases created using publicly available palmprint [2,3] and knuckleprint [1] databases. Experimental result revels that HCF works effectively under environments, having slight amount of variation in illumination, rotation and translation.

Acknowledgements. Authors like to acknowledge the support provided by the Department of Information Technology, Government of India to carry out this research work.

References

1. Knuckleprint Polyu, http://www4.comp.polyu.edu.hk/~biometrics/FKP.htm
2. Palmprint casia, http://www.cbsr.ia.ac.cn
3. Palmprint polyu, http://www.comp.polyu.edu.hk/biometrics
4. Chikkerur, S., Mhatre, A.J., Palla, S., Govindaraju, V.: Efficient search and retrieval in biometric databases. In: Proc. SPIE, vol. 5779, pp. 265–273 (2005)
5. Badrinath, G.S., Nigam, A., Gupta, P.: An efficient finger-knuckle-print based recognition system fusing sift and surf matching scores. In: Qing, S., Susilo, W., Wang, G., Liu, D. (eds.) ICICS 2011. LNCS, vol. 7043, pp. 374–387. Springer, Heidelberg (2011)
6. Badrinath, G.S., Gupta, P.: Palmprint based recognition system using phase-difference information. In: Future Generation Computer Systems (2010) (in Press)
7. Bendale, A., Nigam, A., Prakash, S., Gupta, P.: Iris segmentation using improved hough transform. In: Huang, D.-S., Gupta, P., Zhang, X., Premaratne, P. (eds.) ICIC 2012. CCIS, vol. 304, pp. 408–415. Springer, Heidelberg (2012)
8. Lucas, B.D., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: IJCAI, pp. 674–679 (1981)
9. Meraoumia, A., Chitroub, S., Bouridane, A.: Fusion of finger-knuckle-print and palmprint for an efficient multi-biometric system of person recognition. In: IEEE International Conference on Communications (2011)
10. Meraoumia, A., Chitroub, S., Bouridane, A.: Palmprint and finger knuckle print for efficient person recognition based on log-gabor filter response. Analog Integrated Circuits and Signal Processing 69, 17–27 (2011)
11. Michael, G.K.O., Connie, T., Jin, A.T.B.: An innovative contactless palm print and knuckle print recognition system. Pattern Recognition Letters 31(12), 1708–1719 (2010)
12. Nanni, L., Lumini, A.: A multi-matcher system based on knuckle-based features. Neural Comp. and Appl. 18, 87–91 (2009)
13. Nigam, A., Gupta, P.: Finger knuckleprint based recognition system using feature tracking. In: Sun, Z., Lai, J., Chen, X., Tan, T. (eds.) CCBR 2011. LNCS, vol. 7098, pp. 125–132. Springer, Heidelberg (2011)
14. Nigam, A., Gupta, P.: Comparing human faces using edge weighted dissimilarity measure. In: ICARCV, pp. 1831–1836 (2010)
15. Nigam, A., Gupta, P.: Iris recognition using consistent corner optical flow. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part I. LNCS, vol. 7724, pp. 358–369. Springer, Heidelberg (2013)
16. Zhu, L.Q., Zhang, S.Y.: Multimodal biometric identification system based on finger geometry, knuckle print and palm print. Pattern Recognition Letters 31(12), 1641–1649 (2010)
17. Shi, J., Tomasi, C.: Good features to track. In: Computer Vision and Pattern Recognition, pp. 593–600 (1994)
18. Zhang, L., Zhang, D., Zhu, H.L.: Ensemble of local and global information for finger-knuckle-print recognition. Pattern Recognition 44(9), 1990–1998 (2011)

An Adaptive Comprehensive Learning Bacterial Foraging Optimization for Function Optimization

Lijing Tan[1,*], Hong Wang[2], Xiaoheng Liang[1], and Kangnan Xing[2]

[1] Management School, Jinan University, Guangzhou 510632, China
[2] College of Management, Shenzhen University, Shenzhen 518060, China
mstlj@163.com

Abstract. This paper proposes a variant of the bacterial foraging optimization (BFO) algorithm with time-varying chemotaxis step length and comprehensive learning strategy, namely Adaptive Comprehensive Learning Bacterial Foraging Optimization (ALCBFO). An adaptive non-linearly decreasing modulation model is used to balance the exploration and exploitation. The comprehensive learning mechanism is adopted to maintain the diversity of the bacterial population and thus alleviates the premature convergence. Compared with the classical GA, PSO, the original BFO and two improved BFOs (BFO-LDC and BFO-NDC) , the proposed ACLBFO shows significantly better performance in solving multimodal problems.

Keywords: Bacterial foraging optimization (BFO), adaptive chemotaxis step, comprehensive learning mechanism.

1 Introduction

Bacterial foraging optimization (BFO) was firstly proposed by Passino in 2002 [1]. It inspired by the foraging behavior of *E.coli* bacteria. In the bacterial foraging process, four motile behaviors (chemotaxis, swarming, reproduction, and elimination and dispersal) are mimicked. A brief review of BFO is given in [1].

Recently BFO has been applied successfully to a number of engineering problems, such as economic load dispatch [2], PID controller design [3], portfolio optimization [4] and automated experimental control design [5] etc.

In traditional BFO each individual in the colony independently searches for food by their own experience without any information exchange with others [6]. In dealing with complex problems, BFO has the problems of low convergence speed and easily to be trapped into local minima. To improve its performance, we incorporate an adaptive search mechanism and comprehensive learning strategy to original BFO, called adaptive comprehensive learning bacterial foraging optimization (ACLBFO). Simulation results shown that ACLBFO is expected to enhance alleviate the premature convergence to some extend.

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 194–199, 2013.

The rest of this paper is organized as follows: Section 2 provides a description of ACLBFO algorithm. Experimental settings and results are provided in section 3. Finally, section 4 concludes the paper.

2 Adaptive Comprehensive Learning Bacterial Foraging Optimization Bacterial Foraging Optimization

2.1 Adaptive Mechanism

The chemotaxis step size *C* is constant in the original BFO [1]. However, the previous proposed BFO-LDC [7], BFO-NDC [8] proved that chemotaxis step *C* is a key parameter to keep a right balance between global search and local search. Therefore, the proper selection of size *C* is critical to the success of BFO algorithms.

This work used a non-linearly decreasing modulation model which is based on the local version of adaptive chemotaxis step [9] to optimize the chemotaxis step size. Chemotaxis step size is determined by the following equation:

$$C_j = C_{\min} + \exp(-a * (k/N_{re})^n) * (C_{\max} - C_{\min}) \tag{3}$$

where *n* is the modulation index. *a* is an adjustable coefficient ranged from (0, 7]. N_{re} is the maximum number of bacterial reproduction, *k* is current reproduction time, and C_j is the j^{th} chemotaxis step.

2.2 Comprehensive Learning Mechanism

In this paper, we incorporate comprehensive learning mechanism [10] into BFO, namely Adaptive Comprehensive Learning Bacterial Foraging Optimization (ACLBFO).

In this learning strategy, the moving direction of each bacterium in the d^{th} dimension is updated as follows:

$$\theta_d^i(j+1,k,l) = \theta_d^i(j,k,l) + C(i) * \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} + \lambda * r_1 * (pbest_{id} - \theta_d^i(j,k,l)) + (1-\lambda) * r_2 * (gbest_{id} - \theta_d^i(j,k,l)) \tag{4}$$

$$pbest_{id} = \theta * pbest_{compet} + (1-\theta) * pbest_{id} \tag{5}$$

$$pbest_{compet} = b_i * pbest_n + (1-b_i)\, pbest_m \tag{6}$$

Where: n and m are random individuals belonging to the swarm, $n \in \{1,2,...,S\}$, $m \in \{1,2,...,S\}$, and $n \neq m$. Random numbers: θ and b_i equal to 0 or 1, whilst r_1 and r_2 are fixed values given beforehand. $\lambda \subset \{1,0\}$, and p_c is the learning probability. Specifically, the learning probability of i^{th} bacterium is calculated using the following Equation [11]:

$$p_c^i = \varepsilon + (0.5 - \varepsilon) * \frac{e^{t_i} - e^{t_1}}{e^{t_s} - e^{t_1}} \tag{7}$$

$$\lambda = ceil(rand - 1 + p_c) \tag{8}$$

Where $t_j = 5 * \frac{j-1}{S-1}$, and ε is a fixed parameter given originally. $rand \in [0,1]$. If p_c is larger, then the probability of learning from $pbest_{id}$ is corresponding more, while the learning from $gbest_{id}$ is lesser. Therefore, the bacterium absorbs the information either from individuals' best or the globe group best. Notably, the comprehensive learning mechanism of each bacterium on its moving performs in each dimension rather than the whole dimensions.

The pseudo code of ACLBFO is listed below.

```
Begin
    While (Terminate-condition is not met)
    Evaluate fitness values of the initial population
    Figure out the gbest and the pbest of each bacterium
    For (Elimination-dispersal loop)
       For (Reproduction loop)
          For (Chemotaxis loop)
          Update the chemotaxis step size by Eqn.3
          Compute fitness function
          Update the position by Eqn.4
          Boundary control
          Tumbling, Swimming for N_c steps
          Update the gbest and the pbest
       End For (Chemotaxis loop)
       Compute the health value of each bacterium by Eqn.2
       Sort bacteria based on health values
       Copy the bacteria using health sorting approach
     End For (Reproduction loop)
     Eliminate and disperse according to probability p_ed
   End For (Elimination-dispersal loop)
 End While
End
```

3 Experiments and Results

3.1 Benchmark Functions

To evaluate the effectiveness of the proposed ACLBFO algorithm, two test functions are adopted. The detailed information of the functions, search range *R*, the global optimum *X** and the corresponding fitness *f*(*X**) value of each function are listed in Table 1. n represents the number of dimensions.

Table 1. Parameters of the test functions

Function	Mathematical representation	X^*	$f(X^*)$	R
Sphere	$f_1(x)=\sum_{i=1}^{n} x_i^2$	[0,0,..,0]	0	$[-100, 100]^n$
Griewank	$f_2(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}(\frac{x_i}{\sqrt{i}})+1$	[0,0,..,0]	0	$[-600, 600]^n$

3.2 Parameter Settings for ACLBFO Algorithm

We examined the ACLBFO algorithm by comparing with a standard GA, PSO, the original BFO, and two BFO variants (BFO-LDC and BFO-NDC). The parameters involved in ACLBFO, BFO, BFO-LDC and BFO-NDC are listed in Table 2.

Table 2. Parameter setting for BFO, BFO-LDC, BFO-NDC and ACLBFO

Para. / Alg.	S	N_c	N_s	N_{re}	N_{ed}	P_{ed}	P_r	$C(i)\ (i=1,2,\ldots,S)$
BFO	100	200	5	4	2	0.2	0.5	0.1
BFO-LDC	100	200	5	4	2	0.2	0.5	$C_i=C_{\min}+(\frac{iter_{\max}-iter}{iter_{\max}})*H$
BFO-NDC	100	200	5	4	2	0.2	0.5	$C_i=C_{\min}+\exp(-\lambda(\frac{iter}{iter_{\max}})^2)*H$
ACLBFO	100	200	5	4	2	0.2	0.5	$C_i=C_{\min}+\exp(-\alpha(\frac{k}{N_{re}})^n)*H$

where S is the number of bacteria colony, N_c is Chemotaxis steps, N_s is swimming steps, N_{re} is reproductive steps, N_{ed} is elimination-dispersal steps, P_{ed} is probability of elimination, P_r is probability of keeping, $C(i)$ is the run-length unit, r_1, $r_2(r_1= r_2=0.5)$ are two fixed parameters impacting the chemotaxis direction θ, $P_c(=0.1)$ is the probability of learning, $n(=40)$ is the number of dimensions, $H=C_{\max}-C_{\min}$.

GA settings: We fixed the population size at 100 and other parameters were set as indicated in [11]. PSO settings: The acceleration factors $C_1=C_2=2.0$. Inertia weight was decayed from 0.9 to the end 0.6 as recommended in [12].

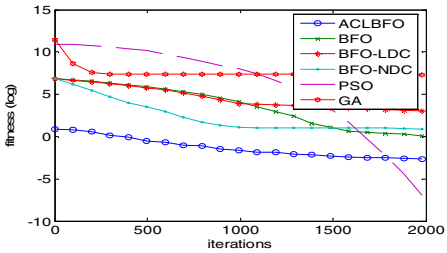
3.3 Simulation Results and Performance Assessment

Due to the page limitations, we only present the experimental results for two selected benchmark functions, which are listed in Table 3. The results shown in bold are the best results of the six algorithms in each function.

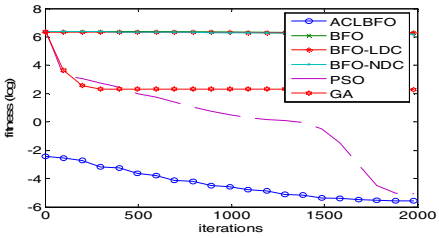
From Table 3, we observe that ACLBFO outperforms other algorithms on the multimodal problem, However, do not perform best on Sphere function. Due to the comprehensive learning strategy, each bacterium can learn information not only from itself but also from all other bacteria's historical experience. We can conclude that the ACLBFO is more effective in solving problems with less linkage (base on the characteristic of Griewank function).

Table 3. Experimental results on benchmark functions (40-D)

Func / Alg.	Unimodal f_1(Sphere)	Multimodal f_2(Griewank)
GA	1.5075e+003 ± 649.0831	9.9175 ± 3.4341
PSO	**6.9008e-004 ± 8.5408e-004**	0.0062 ± 0.0099
BFO	2.3968 ± 0.2435	534.8022 ± 74.7301
BFO-LDC	19.7817 ± 3.5696	546.2333 ± 62.7786
BFO-NDC	2.3968 ± 0.2435	481.7740 ± 35.3368
ACLBFO	0.0832 ± 0.0313	**0.0030 ± 9.9001e-004**



(a).Sphere function



(b). Griewank function

Fig. 1. The median convergence characteristics of 40-D test functions

Fig.1 shows the comparison of the convergence curves of ACLBFO, GA, PSO, BFO, and other two improved BFO algorithms, conducted on the two benchmark functions. It can be seen that the convergence behavior of the ACLBFO in Griewank function has been well improved in comparison to its counterparts. In Sphere function, ACLPSO performs better at the beginning but finally be outperformed by PSO.

4 Conclusions and Future Work

This paper proposed a novel Adaptive Comprehensive Learning Bacterial Foraging Optimization (ACLBFO). Experiment results have proved that ACLBFO algorithm was more effective in finding better solutions in multimodal problems. But in unimodal function, it is not the best performed algorithm. This is just what we should

focus in the further work. Last but not least, it is desired to be tested further and applied to solve those more comprehensive real-world problems.

Acknowledgements. This work is supported by National Natural Science Foundation of China (Grant Nos.71001072, 71271140, 71210107016, 71240015), and The Natural Science Foundation of Guangdong Province (Grant nos. S2012010008668, 9451806001002294).

References

1. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine, 52–67 (2002)
2. Panigrahi, B.K., Pandi, V.R.: Bacterial foraging optimization: Nelder-Mead hybrid algorithm for economic load dispatch. IET Generation Transmission & Distribution 2(4), 556–565 (2008)
3. Kou, P.G., Zhou, J.Z., He, Y.Y., Xiang, X.Q., Li, C.S.: Optimal PID governor tuning of hydraulic turbine generators with bacterial foraging particle swarm optimization algorithm. Proceedings of the Chinese Society of Electrical Engineering 29(26), 101–106 (2009)
4. Niu, B., Fan, Y., Xiao, H., Xue, B.: Bacterial foraging based approaches to portfolio optimization with liquidity risk. Neurocomputing 98(3), 90–100 (2012)
5. Okaeme, N.A., Zanchetta, P.: Hybrid bacterial foraging optimization strategy for automated experimental control design in electrical drives. IEEE Transactions on Industrial Informatics 9(2), 668–678 (2013)
6. Niu, B., Wang, H.: Bacterial colony optimization. Discrete Dynamics in Nature and Society, 1–28 (2012)
7. Niu, B., Fan, Y., Zhao, P., Xue, B., Li, L., Chai, Y.J.: A novel bacterial foraging optimizer with linear decreasing chemotaxis step. In: 2nd International Workshop on Intelligent Systems and Applications (ISA), pp. 1–4 (2010)
8. Niu, B., Fan, Y., Wang, H.: Novel bacterial foraging optimization with time-varying chemotaxis step. International Journal of Artificial Intelligence, 257–273 (2011)
9. Niu, B., Wang, H., Tan, L.J., Li, L.: Improved BFO with adaptive chemotaxis step for global optimization. In: International Conference on Computational Intelligence and Security (CIS), pp. 76–80 (2011)
10. Liang, J.J., Qin, A.K., Suganthan, P.N.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation 10(3), 281–295 (2006)
11. Ashlock, D.: Evolutionary computation for modeling and optimization. Springer, New York (2006)
12. Shi, Y., Eberhart, R.C.: Empirical study of particle swarm optimization. In: Proceedings of the IEEE Congress Evolutionary Computation, pp. 1945–1950 (1999)

A Multi-objective Particle Swarm Optimization Based on Decomposition

Yanmin Liu[1,2,*] and Ben Niu[3,*]

[1] School of mathematics and computer science, Zunyi Normal College, Zunyi 563002
[2] School of economics and management, Tongji University, Shanghai, 200092, China
[3] College of Management, Shenzhen University, Shenzhen, 518060
{drniuben,yanmin7813}@gmail.com

Abstract. Decomposition is a classic method in traditional multi-objective optimization problems (MOPs). However, so far it has not yet been widely used in multi-objective particle swarm optimization (MOPSO). This paper proposes a MOPSO based on decomposition strategy (MOPSO-D), in which MOPs is decomposed into a number of scalar optimization sub-problems by a set of even spread weight vectors, and each sub-problem is optimized by a particle (here, it is viewed as a sub-swarm) personal history best position (*pbest*) and global best position in the its all neighbors (*gbest*) in a single run. By computing the Euclidean distances between any two weight vectors corresponding to a particle, the neighborhood identification strategy of each particle is assigned. The method of decomposition inherited the traditional method merits and makes MOPSO-D have lower computational complexity at each generation than NSMOPSO and OMOPSO. Simulation experiments on multi-objective 0-1 knapsack problems and continuous multi-objective optimization problems show MOPSO-D outperforms or performs similarly to NSMOPSO and OMOPSO.

Keywords: Particle swarm optimizer, Decomposition, Multi-objective optimization problems.

1 Introduction

The use of evolutionary algorithms for dealing with multi-objective optimization has a growing interest in the last decade, giving rise to a wide variety of algorithms [1][2]. Particle swarm optimization (PSO) [3] is a relatively recent heuristic inspired by the choreography of a bird flock, which makes PSO particularly suitable for multi-objective optimization (MOPs) because of the high speed of convergence presented in scalar objective optimization.

In the domain of extending particle swarm optimization to handle multi-objective optimization problems, there is no decomposition method involved in the majority of the current state-of-the-art MOPSO [1-6], in which they deal with multiple objection as a whole, and there is no association in each individual solution with any particular

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 200–205, 2013.

scalar optimization problem. It was well known that the solutions can be compared by objective function values, and the best solution (just one) can be found in PSO run in scalar objective optimization. However, in MOPs, Pareto domination relation does not give a complete ordering among the solutions in the objective space. Accordingly, the aim of MOPSO run is to generate a number of Pareto optimal solutions based on three common goals, i.e. maximizing the number of elements of the Pareto optimal set found, minimizing the distance of the Pareto front produced by an algorithm with respect to the true Pareto front, and maximizing the distribution of solutions found. In order to achieve this goal, decomposition method has been used to a certain extent in several meta-heuristics [7], and it has been proven to be effective in dealing MOPs.

This paper is organized as follows: In Section 2, the proposed algorithm MOPSO-D is elaborated. Comparative study and pertinent discussion are presented in Section 3. Finally, Section 4 provides concluding remarks of this paper.

2 MOPSO Based on Decomposition (MOPSO-D)

2.1 Decomposition Method

There are several approaches for converting the problem of approximation of the PF into a number of scalar optimization problems, such as weighted sum approach, Tchebycheff approach [7] and so on. In this paper, we only introduce the adopted Tchebycheff method, and other approaches are presented in [7].

In the Tchebycheff approach, the decomposed scalar optimization problem is the following,

$$\begin{aligned} &\min g^{te}(x \mid \lambda, z^*) = \max_{1\le i\le m}\{\lambda_i \mid f_i(x) - z_i^* \mid\} \\ &\text{subject to } x \in \Omega \end{aligned} \tag{1}$$

where $z^* = (z_1^*, z_2^*, \cdots, z_m^*)^T$ is the reference point, i.e., $z_i^* = \min\{f_i(x) \mid x \in \Omega\}$ for $i = 1,2,\cdots,m$.

2.2 MOPSO-D General Framework

Let $\lambda_1, \lambda_2, \cdots, \lambda_n$ be a set of even spread weight vectors and z^* be the reference point, so, the problem of approximation of the PF of (1) can be decomposed into *N* scalar optimization sub-problems (Note that a sub-problem is regard as a sub-swarm, which includes a particle in this paper) by using the Tchebycheff approach and the objective function of the j*th* sub-problem is

$$\min g^{te}(x \mid \lambda^j, z^*) = \max_{1\le i\le m}\{\lambda_i^j \mid f_i(x) - z_i^* \mid\} \tag{2}$$

where $\lambda^j = (\lambda_1^j, \lambda_2^j, \cdots, \lambda_m^j)$, MOPSO-D minimizes all *N* objection functions simultaneously in a evolution run. MOPSO-D pseudo-code is described in Table 1. It should be noted that the improved PSO proposed by [8] is adopted in this paper.

Table 1. Psedudo-code of MOPSO-D

MOPSO-D pseudo-code

Initialization PSO parameter:

- Max iteration, acceleration constant c_1 and c_2 , inertia weight
- N: the number of the sub-problems considered in MOPSO-D
- A uniform spread of N weight vectors: $\lambda_1, \lambda_2, \cdots, \lambda_n$;
- T: the number of the weight vectors in the neighborhood of each weight vector.
- EP: the external archive for storing the Non-dominated and reporting results.

Optimization:

Step 1: Update

Step 1.1 Set $EP = \phi$.

Step 1.2 Compute the Euclidean distances between any two weight vectors and then work out the T closest weight vectors to each weight vector (corresponding to a particle).For $i = 1,2,\cdots,N$, set $B(i) = \{i_1, i_2, \cdots, i_T\}$, where $\lambda^{i_1}, \lambda^{i_2}, \cdots, \lambda^{i_T}$ are the T closest weight vectors to λ^i .

Step 1.3 Generate an initial population $x^1, x^2, \cdots, x^N$ in terms of the variable domain.

Step 1.4 Initialize $z = (z_1, z_2, \cdots, z_m)^T$ by $z_i^* = \min\{f_i(x) \mid x \in \Omega\}$.

Step 2: Main loop (For $i = 1,2,\cdots,N$)

Step 2.1 Reproduction: generate a new solution x by PSO evolution equation (3) and (4).

Step 2.2 Update z: For each $j = 1,2,\cdots,m$, if $z_j < f_j(y)$, then set $z_j = f_j(y)$.

Step 2.4 Update $pbest$ and $gbest$ in terms of $g^{te}(pbest \mid \lambda^j, z)$, $g^{te}(x \mid \lambda^j, z)$

Step 2.5 Update EP.

Step 3: Stopping Criteria: If stopping criteria is satisfied

Report results

3 Simulation Experiment and Analysis

3.1 Performance Evaluation

Both quantitative and qualitative comparisons are made to validate the EPMOPSO algorithm against other MOPSOs. For qualitative comparison, the plots of final Pareto fronts are presented. As for the quantitative comparison, convergence metric (γ) and spread (Δ) [1] are used, which is shown in Eqn (3) and (4).

$$\gamma = \frac{\sqrt{\sum_{i=1}^{N} d_i^2}}{N} \tag{3}$$

$$\Delta = \frac{\sum_{m=1}^{M} d_m^{\,e} + \sum_{i=1}^{N-1} \left| d_i - \overline{d} \right|}{\sum_{m=1}^{M} d_m^{\,e} + (N-1)\overline{d}} \tag{4}$$

where the parameter d_m^e is the Euclidean distance between the extreme solutions of Pareto optimal front and the boundary solutions of the obtained non-dominated set corresponding to the m^{th} objective function, the parameter d_i is the Euclidean distance between neighboring solutions in the obtained non-dominated solutions set, and $\overline{d}$ is the mean value of these distances. The smaller the value of Δ means the better the diversity of the non-dominated set.

3.2 Test Functions

To illustrate the efficiency of the proposed MOPSO-D algorithm tow benchmark problems are selected, i.e., multi-objective 0-1 knapsack problem [9]and continuous multi-objective optimization problems (ZDT1)[2].

3.3 Results and Discussion

In order to know how competitive MOPSO-D was, we compare it with two multi-objective PSO algorithms that are representative of the state of the art. These two algorithms are NSMOPSO [4], OMOPSO [10]. Each algorithm is run 30 times to

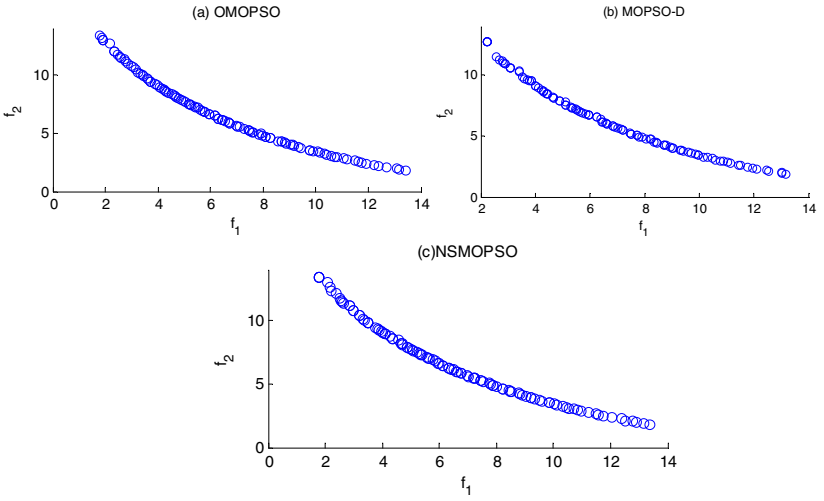


Fig. 1. Non-dominated fronts in kno1

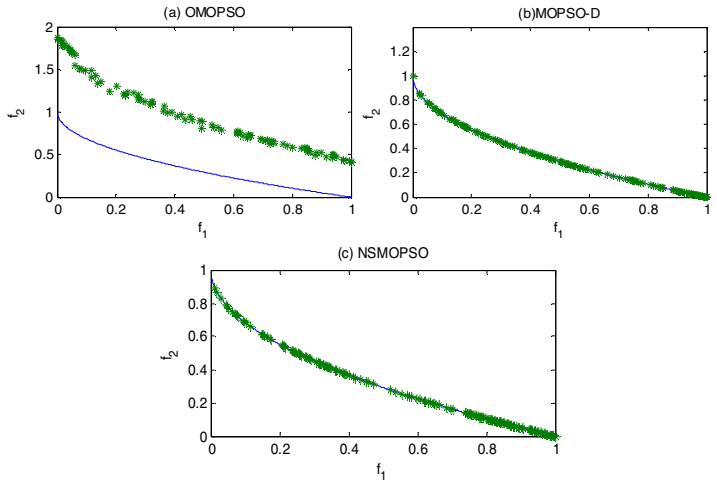


Fig. 2. Non-dominated fronts in ZDT1

achieve metric (γ) and (Δ) for each test function. The mean values and standard deviation of the results are collected in Tables 1. The resulting non-dominated fronts are plotted in Figures 1 and 2.

Table 2. Comparison of performance on ZDT1

	NSMOPSO		OMOPSO		MOPSO-D	
	γ	Δ	γ	Δ	γ	Δ
Best	0.088	0.69	0.084	0.67	0.0027	0.33
Worst	0.635	0.87	0.123	0.98	0.0098	0.78
Average	0.195	0.75	0.084	0.57	0.0059	0.65
Median	0.188	0.78	0.083	0.56	0.0067	0.71
Variance	9.25e-04	8.29e-04	5.98e-05	5.59e-04	5.53e-06	9.95e-05

From non-dominated fronts (Figure 1 and 2) and Table 1, we can find that MOPSO-D is able to find the well-distributed and near-optimal Pareto fronts for all test functions compared with other two algorithms. Therefore, MOPSO-D may be used as a method to solve multi-objective objection.

4 Conclusions

This paper proposes a MOPSO based on decomposition strategy (MOPSO-D), in which MOPs is decomposed into a number of scalar optimization sub-problems, and each sub-problem is optimized by only using information from its several neighboring sub-problems in a single run. Both two performance metrics (γ and Δ), it clearly indicate that MOPSO-D is highly competitive and even outperforms the selected

MOPSOs. The figures of Pareto fronts also show that MOPSO-D has the ability to produce relatively better-distributed Pareto fronts compared with the selected MOPSOs.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (Grants nos. 71001072, 71271140, 71210107016, 71240015), and the Natural Science Foundation of Guangdong Province (Grant nos. S2012010008668, 9451806001002294). Guizhou province science and technology fund (Qian Ke He J [2012] 2340, LKZS [2012]10), China Postdoctoral Science Foundation Funded Project (Grant no. 2012M520936), Shanghai Postdoctoral Science Foundation Funded Project (Grant no. 12R21416000), Guizhou province college outstanding scientific and technological innovation talent support projects (Qian Jiao HE KY[2012]104).

References

1. Coello, C.A.C., Lechuga, M. S.: MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization. In: IEEE Congress on Evolutionary Computation, Piscataway, New Jersey, pp. 1051–1056. IEEE Press, New York (2002)
2. Hu, X., Eberhart, R. C.: Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization. In: Proceedings of the 2002 Congress on Evolutionary, Honolulu, HI, pp. 1677–1681. IEEE Press, New York (2002)
3. Kennedy, J., Eberhart, R. C.: Particle Swarm Optimization, in: Proceedings of the 2002 Conference on Neural Networks, Piscataway, pp. 1942-1948. IEEE Press, New York (1995)
4. Li, X.: A Non-dominated Sorting Particle Swarm Optimizer for Multi-objective Optimization. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 37–48. Springer, Heidelberg (2003)
5. Mostaghim, S., Teich, J.: Strategies for Finding Good Local Guides in Multi-objective Particle Swarm Optimization (MOPSO). In: IEEE Swarm Intelligence Symposium, Indianapolis, pp. 26–33. IEEE Press, New York (2003)
6. Mostaghim, S., Teich, J.: The Role of ε-dominance in Multi-Objective Particle Swarm Optimization Methods. In: IEEE Congress on Evolutionary Computation, Canberra, Australia, pp. 1764-1771. IEEE Press, New York (2003)
7. Jin, Y., Okabe, T., Sendhoff, B.: Adapting Weighted Aggregation for Multi-objective Evolutionary Strategies in Evolutionary Multicriterion Optimization. In: Zitzler, E. (Eds.) EMO 2001(LNCS). vol. 1993, pp. 96–110. Springer, Heidelberg (2001)
8. Liu Y. M. Niu B.: A Novel PSO Model Based on Simulating Human Social Communication Behavior. Discrete Dynamics in Nature and Society, 1-21(2012)
9. Bazgan C., Hugot H., Vanderpooten D.: Solving Efficiently the 0-1 Multi-objective Knapsack Problem, 36, 260-279 (2009).
10. Hughes, E. J.: Multiple single objective Pareto sampling. In: IEEE Congress on Evolutionary Computation, Canberra, Australia, pp. 2678–2684. IEEE Press, New York (2003)

Consensus of Sample-Balanced Classifiers for Identifying Ligand-Binding Residue by Co-evolutionary Physicochemical Characteristics of Amino Acids*

Peng Chen

Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia
peng.chen@kaust.edu.sa

Abstract. Protein-ligand binding is an important mechanism for some proteins to perform their functions, and those binding sites are the residues of proteins that physically bind to ligands. So far, the state-of-the-art methods search for similar, known structures of the query and predict the binding sites based on the solved structures. However, such structural information is not commonly available. In this paper, we propose a sequence-based approach to identify protein-ligand binding residues. Due to the highly imbalanced samples between the ligand-binding sites and non ligand-binding sites, we constructed several balanced data sets, for each of which a random forest (RF)-based classifier was trained. The ensemble of these RF classifiers formed a sequence-based protein-ligand binding site predictor. Experimental results on CASP9 targets demonstrated that our method compared favorably with the state-of-the-art.

Keywords: Protein-ligand binding, Random forest, Co-evolutionary encoding.

1 Introduction

Protein-ligand binding is an important mechanism for some proteins to perform their functions. Protein-ligand binding sites are the residues of proteins that physically bind to ligands. In biochemistry, a ligand usually is a small molecule that always forms a complex with a molecule to serve a specific biological purpose. Commonly, there are several ligand categories: "metal ions" (e.g., Ca), "inorganic anions" (e.g., SO4), "DNA/RNA" for poly-ribonucleic acid binding sites, and "organic ligands" for cofactors, substrates, and receptor agonists/antagonists (e.g., NAD) [21], and so on.

A number of methods applied nuclear magnetic resonance (NMR) spectroscopy [20, 2, 12, 3, 13, 16, 1] or X-ray [18] to determining protein structures. Such structural information is essential to determine the ligand-binding sites. Pintacuda *et al.* employed lanthanide ions for the determination of protein-ligand binding sites [20].

* This work was supported Award Numbers KUS-CI-016-04 and GRP-CF-2011-19-P-Gao-Huang, made by King Abdullah University of Science and Technology (KAUST).

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 206–212, 2013.

Since experimental efforts to determine ligand-binding sites are always time-consuming, computational methods are needed that can assist in the identification of such sites.

Most computational approaches searched for similar or homologous structures of the query sequence to determine its ligand-binding sites from the homologous structures [21, 11]. Those structure-based techniques are restricted by the limited number of available protein structures. Therefore, sequence-based approaches are useful when structural information is not available. A number of sequence-based methods have been developed to predict ligand-binding states. For example, Passerini and co-workers introduced a method for identifying states of histidines and cysteines [19].

However, the problem of whether ligand-binding sites can be predicted from sequence information remains open. Little progress has been made on the sequence-based ligand-binding site prediction. Kauffman and Karypis proposed a method that combined machine learning and homology information for the sequence-based ligand-binding site prediction [14]. However, the method did not perform well. In this paper, we propose a sequence-based approach, BindRFs (Binding site prediction by the ensemble of Random Forest classifiers), to identify protein ligand-binding residues based on the co-evolutionary context of amino acid residues. First, due to the imbalanced samples between ligand-binding sites and non-binding sites, several data sets are constructed. Each of them is composed of the binding site subset (positive subset) and part of the non-binding site subset (negative subsets). All the negative subsets are disjoint with each other. Then a random forest (RF) classifier is learned on each data set. Experiments on the CASP9 data set show that the consensus of these RF classifiers can yield good prediction on ligand-binding sites.

2 Methods

2.1 Feature Vector Representation of a Residue

In the AAindex1 database [15], there are 544 amino acid (AA) properties. Many of these properties are correlated. We thus extracted relatively irrelevant properties with a correlation coefficient (CC) of 0.8. For each property, the CC to all the other properties was calculated and the number of related properties was counted. The 544 properties were thus ranked by their numbers of related properties. From the top property, we removed from the list all the properties that were related to it. This was repeatedly done until no related pair existed in the list, which resulted in 174 properties.

For a residue i in a protein chain, the association among the neighboring residues is considered in this work. A sliding window that contains seven residues centered at the residue i is used to encode the features. An encoding schema integrating AA properties and sequence profile is used to represent the residue. The sequence profile for one residue created by PSI-Blast with default parameters [4] is then multiplied by each AA property. For instance, the profile SP^k, k=1,…,7 for residue k in the seven residue window and one AA property scale, AAP_j, are both vectors with 1×20 dimensions. Thereafter, $MSK^k_j=SP^k\times AAP_j$ for residue k represents the multiplication

of the corresponding sequence profile by the scale, where $\times$ represents the element-wise product. In our previous work, we found out that the standard deviation of MSK^k_j reflected the evolutionary variance of the residue k along with the AA property AAP_j [6, 8, 9].

2.2 Ensemble of Random Forest Classifiers

Machine learning techniques have played very important role in various protein-related problems, such as function annotation [16], membrane protein type prediction [23], and protein retrieval [24]. Here we propose to use the random forest model, which consists of an ensemble of simple tree predictors, each of which depends on a set of random features selected independently [5]. For the ligand-binding site prediction problem, the ensemble of simple trees votes for the most popular ligand-binding site class. Previous results showed that using consensus ideas can make significant improvement in prediction accuracy [10, 7].

Given a set of training data $D_N=\{(X_i, Y_i)\}$, $i=1,\ldots,N$, let the number of training instances be N, the number of features in the classifier be J, and the number of trees to build be K. For each tree, a number of j features is considered to determine the decision of the tree, where j should be much less than J and set as $1 \leq j \leq int(log(J)+1)$ by default. For the k^{th} tree, a random vector v_k is generated, which is independent and with the same distribution of the previous ones, $v_1,\ldots,v_{k-1}$. The k^{th} tree generated from the training set x and v_k results in a classifier $CF_k(x; v_k)$, where $k=1,\ldots,K$.

After all of the trees are generated, for a query instance X, they vote for the most popular class and thus the prediction of the whole forest is,

$$F(X)=majority\ vote\ \{CF_k(X)\}^K_{k=1}. \quad (1)$$

Since the binding site data set is highly imbalanced, i.e., only 3.9% of all the instances are positive samples, balancing the positive (binding site class) and the negative (non-binding site class) data is necessary to avoid the overfitting of classifiers. Thus, 25 data sets are formed, D^n_N, $n=1,\ldots,25$, and they share the same positive data, but have disjoint negative data. A random forest classifier is trained for each of the 25 data sets. The final prediction is the majority voting of the 25 random forests.

2.3 Datasets, Binding Site Definition, and Evaluation Criteria

We took the targets in the CASP9 competition as our ligand-binding site data set, where there were 30 targets with bound ligands [21].

For each protein, all residues, at least one heavy atom within a given distance from any heavy atom of the ligand, were defined as ligand-binding site residues. In fact, the definition of the distance cutoff was different in literature. Kauffman and Karypis collected ligand-binding residues having at least one heavy atom within 5 Å of a ligand [14]. While in CASP9, the distance cutoff was defined as the sum of the van der Waals radii of the involved atoms plus a tolerance of 0.5 Å. Different distance cutoff leads to different ligand-binding site data set, i.e., about 9% of residues are the ligand-binding residues for the former definition, while only 3.9% for the latter.

In this work we adopted five evaluation measures to evaluate the performance of our method, i.e., sensitivity (Sen), precision (Prec), F-measure (F1), Matthews correlation coefficient (MCC) [6, 22], and the area (AUC) under the receiver operating characteristic (ROC) curve.

3 Results

We first analyzed the AA preferences for ligand binding sites and non-ligand binding sites. Results show that amino acids Asp, Gly and His frequently occur in the ligand binding sites, while amino acids Leu and Ala are often regarded as non-ligand binding sites. However, it may not always be the case because our current data set is relatively small. Despite of that, Asp and His are always considered as hydrophilic amino acids while Leu and Ala as hydrophobic ones, which is partially in agreement with our statistics. In fact, hydrophilic amino acids seem to be more likely to be ligand binding sites.

In this work, we used a 10 fold cross-validation to evaluate our method, i.e., instances of the data set were roughly divided into 10 subsets and in turn, our model was trained on nine of them and test on the remaining one. Table 1 shows the performance comparison of the 25 RF classifiers and that of the ensemble. From Table 1, it can be seen that the ensemble of the 25 RF classifiers with majority voting performs well. It yields the MCC of 0.33 and the F1-score of 0.35, which significantly outperforms any individual RF classifier, where the best individual prediction, the 18th classifier, achieves the MCC of 0.23 and an F1 of 0.26.

Table 1. Overall prediction performance of the 25 RF classifiers and that of the.

Classifier	No.	Sen	MCC	Prec	F1	No.	Sen	MCC	Prec	F1
	1	0.25	0.22	0.26	0.25	14	0.27	0.23	0.25	0.26
	2	0.25	0.20	0.23	0.24	15	0.28	0.19	0.19	0.23
	3	0.56	0.18	0.11	0.19	16	0.45	0.22	0.16	0.23
	4	0.44	0.20	0.15	0.22	17	0.43	0.19	0.14	0.21
	5	0.50	0.21	0.14	0.21	**18**	**0.39**	**0.23**	**0.19**	**0.26**
	6	0.44	0.21	0.16	0.23	19	0.23	0.21	0.24	0.24
Individual	7	0.21	0.20	0.24	0.22	20	0.43	0.21	0.16	0.23
	8	0.52	0.21	0.14	0.22	21	0.38	0.21	0.17	0.24
	9	0.40	0.22	0.18	0.25	22	0.38	0.22	0.18	0.24
	10	0.37	0.22	0.19	0.25	23	0.49	0.20	0.14	0.21
	11	0.19	0.22	0.31	0.24	24	0.43	0.20	0.15	0.22
	12	0.36	0.22	0.19	0.25	25	0.43	0.21	0.15	0.23
	13	0.51	0.22	0.15	0.23					
Ensemble		0.31	0.33	0.41	0.35					

It is difficult to compare our model with other similar methods for there is no method using only the sequence information to predict ligand binding sites. Most of ligand binding site prediction methods applied structural information of homologous proteins in the prediction. LIBRUS's model only using sequence information can achieve a PR-AUC of 0.29. For comparison, our method can also achieve a PR-AUC

of 0.30, but our used CASP9 data set containing less ligand binding sites for each protein. In CASP9 meeting, CASP9 abstract reported that LIBRUS's performance was ROC-AUC=0.66 and precision/recall=0.24 which corresponds to 15% precision at 50% recall when combining sequence information and homology-based transfer. Our method yields a ROC-AUC of 0.80 and a precision of 41% at 30% recall by only using sequence information. And our method obtains an F1 of 0.35 which is larger than the LIBRUS's F1 of 0.23. Moreover, the random predictor is also implemented here and ran 100 times. The average performance is appended in the end of Table 2. Results show that our method outperforms the random predictor by 36 times of the MCC score and 6 times of the F1 score.

Table 2. Performance comparison of the three methods on CASP9 data set

Method	Type	Sen	MCC	Prec	F1
BindRFs	Random Forest	0.30	0.33	0.41	0.35
LIBRUS	SVM	0.50	-	0.15	0.23
Random Predictor		0.10	0.01	0.05	0.06

Although it is difficult to identify ligand binding sites, our method yields a good result based on sequence information only. Our method obtains an area under the PR curve (PR) of 0.30 while the random predictor only has a PR-AUC of 0.04.

4 Conclusions

This paper proposes an ensemble of RF classifiers with only sequence information to predict ligand binding sites. In order to balance the ligand site data set, several data sets are constructed and composed of the binding site subset (positive subset) and one of the non-binding site subsets (negative subsets), each of the negative subsets are independent with the others. Then each data set is input into a RF classifier. The ensemble of these RF classifiers can yield good prediction on ligand-binding sites. The encoding schema integrating those properties and evolutionary information of amino acids is important to obtain the evolutionary context of ligand binding site residues and thus, the method yields good performance on predicting ligand binding sites. Although structure-based methods still outperform sequence-based methods, our method provides a potential alternative solution to the binding site prediction problem, especially when structure information is not available.

References

1. Abbas, A., Kong, X.B., Liu, Z., et al.: Automatic Peak Selection by Abenjamini-hochberg-based Algorithm. PLoS One 8(1), e53112 (2013)
2. Alipanahi, B., Gao, X., Karakoc, E., et al.: Picky: A Novel Svd-based Nmr Spectra Peak Picking Pethod. Bioinformatics 25(12), i268–i275 (2009)

3. Alipanahi, B., Gao, X., Karakoc, E., et al.: Error Tolerant Nmr Backbone Resonance Assignment and Automated Structure Generation. J. Bioinform. Comput. Biol. 9(1), 15–41 (2011)
4. Altschul, S.F., Madden, T.L., Schaffer, A.A., et al.: Gapped Blast and Psi-blast: A New Generation of Protein Database Search Programs. Nucleic Acids Res. 25(17), 3389–3402 (1997)
5. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)
6. Chen, P., Li, J.: Sequence-based Identification of Interface Residues by An Integrative Profile Combining Hydrophobic and Evolutionary Information. BMC Bioinformatics 11, 402 (2010)
7. Chen, P., Li, J.: Prediction of Protein Long-range Contacts Using An Ensemble of Genetic Algorithm Classifiers with Sequence Profile Centers. BMC Struct. Biol. 10(Suppl. 1), S2 (2010)
8. Chen, P., Wong, L., Li, J.: Detection of Outlier Residues for Improving Interface Prediction in Protein Heterocomplexes. IEEE/ACM Trans. Comput. Biol. Bioinform. 9(4), 1155–1165 (2012)
9. Chen, P., Li, J., Wong, L., et al.: Accurate Prediction of Hot Spot Residues Through Physicochemical Characteristics of Amino Acid Sequences. Proteins (2013)
10. Gao, X., Bu, D., Xu, J., et al.: Improving Consensus Contact Prediction via Server Correlation Reduction. BMC Struct. Biol. 9, 28 (2009)
11. Gonzalez, A.J., Liao, L., Wu, C.H.: Predicting ligand binding residues and functional sites using multipositional correlations with graph theoretic clustering and kernel cca. IEEE/ACM Trans. Comput. Biol. Bioinform. 9(4), 992–1001 (2012)
12. Jang, R., Gao, X., Li, M.: Towards Fully Automated Structure-based NMR Resonance Assignment of 15N-labeled Proteins from Automatically Picked Peaks. J. Comput. Biol. 18(3), 347–363 (2011)
13. Jang, R., Gao, X., Li, M.: Combining automated peak tracking in SAR by NMR with structure-based backbone assignment from 15N-NOESY. BMC Bioinformatics 13(Suppl. 3), S4 (2012)
14. Kauffman, C., Karypis, G.: Librus: Combined Machine Learning and Homology Information for Sequence-based Ligand-binding Residue Prediction. Bioinformatics 25(23), 3099–3107 (2009)
15. Kawashima, S., Pokarowski, P., Pokarowska, M., et al.: Aaindex: Amino Acid Index Database, Progress report 2008. Nucleic Acids Res. 36(Database issue), D202–D205 (2008)
16. Liu, Z., Abbas, A., Jing, B.Y., et al.: Wavpeak: Picking Nmr Peaks Through Wavelet-Based Smoothing and Volume-based Filtering. Bioinformatics 28(7), 914–920 (2012)
17. Messih, M.A., Chitale, M., Bajic, V.B., et al.: Protein Domain Recurrence and Order Can Enhance Prediction of Protein Functions. Bioinformatics 28(18), i444–i450 (2012)
18. Palmer, R.A., Niwa, H.: X-ray Crystallographic Studies of Protein-ligand Interactions. Biochem. Soc. Trans. 31(Pt. 5), 973–979 (2003)
19. Passerini, A., Punta, M., Ceroni, A., et al.: Identifying Cysteines and Histidines in Transition-metal-binding Sites Using Support Vector Machines and Neural Networks. Proteins 65(2), 305–316 (2006)
20. Pintacuda, G., John, M., Su, X.C., et al.: Nmr Structure Determination of Protein-Ligand Complexes by Lanthanide Labeling. Acc. Chem. Res. 40(3), 206–212 (2007)
21. Schmidt, T., Haas, J., Gallo Cassarino, T., et al.: Assessment of Ligand-binding Residue Predictions in Casp9. Proteins 79(Suppl. 10), 126–136 (2011)

22. Wang, B., Chen, P., Huang, D.S., et al.: Predicting Protein Interaction Sites from Residue Spatial Sequence Profile and Evolution Rate. FEBS Lett. 580(2), 380–384 (2006)
23. Wang, J., Li, Y., Wang, Q., et al.: Proclusensem: Predicting Membrane Protein Types by Fusing Different Modes of Pseudo Amino Acid Composition. Comput. Biol. Med. 42(5), 564–574 (2012)
24. Wang, J., Gao, X., Wang, Q., et al.: Prodis-contshc: Learning Protein Dissimilarity Measures and Hierarchical Context Coherently for Protein-protein Comparison in Protein Database Retrieval. BMC Bioinformatics 13(Suppl. 7), S2 (2012)

An Adaptive Approach for Content Based Image Retrieval Using Gaussian Firefly Algorithm

T. Kanimozhi[1] and K. Latha[2]

[1] Dept. of Computer Science & Engg, BIT Campus, Anna University, Tiruchirappalli, Tamilnadu, India
csrkani@gmail.com
[2] Dept. of Computer Science & Engg, BIT Campus, Anna University, Tiruchirappalli, Tamilnadu, India
erklatha@gmail.com

Abstract. An adaptive content based image retrieval (CBIR) approach based on relevance feedback and Gaussian Firefly algorithm is proposed in this paper. Feature extraction has been done with the Euclidean distance estimation between the pixels; relevance feedback (RF) based approach but all concerns with the extraction of image accuracy. This research work has a focused approach to increase the performance by optimizing image feature by adopting with the firefly algorithm (FA). Further, to improve the retrieval accuracy, random walk concepts based on Gaussian distribution is used to move all the fireflies to global best at the end of each iteration. Experiments demonstrate that the proposed method shows more accuracy and better performance compared to particle swarm optimization and genetic algorithm and the use of Gaussian distribution further improve the retrieval accuracy.

Keywords: Content-based image retrieval, Relevance Feedback, Firefly Algorithm, color descriptor, texture descriptor, Gaussian distribution.

1 Introduction

With the huge requirements of multimedia information processing, multimedia information retrieval becomes essential, among which image retrieval has becoming widely recognized. Particularly, content-based image retrieval [1], [2] has gained much attention in the past decades. CBIR is a technique to retrieve images from an image database such that the retrieved images are semantically relevant to a query image provided by a user based on low-level visual features which is still far from satisfactory.

Relevance feedback (RF) has been demonstrated to be a powerful tool which involves the user in the loop to enhance the performance of CBIR [3], [4]. But RF approaches also having the critical issues like long iteration problem. To encounter the issues in relevance feedback of the image retrieval system, we considered the

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 213–218, 2013.

speculative and effective design in which the RF technique is integrated into meta-heuristics firefly. Recently, a new modern meta-heuristic algorithm, called firefly algorithm, developed by Xin-she Yang [5] is a population based technique. This algorithm mimics some of the characteristics of firefly swarms and their flashing behavior. A firefly with lower flash intensity tends to be attracted towards other fireflies with higher flash intensity in which the light intensity decreases as the distance increases. This algorithm has two advantages which are local attractions and automatic regrouping and is thus different from particle swarm optimization. This later advantage makes it suitable for global optimization problems.

But firefly algorithm has some disadvantage such as the parameters of this algorithm are fixed and do not change by the time. Firefly algorithm sometimes traps into several local optimum solutions and can't recover from them. Thus, in this paper random walk concepts based on Gaussian distribution [6] is used in firefly algorithm to move all fireflies to global best solutions at the end of each iteration which overcomes the above disadvantages. All the works are progressed through color and texture features.

The rest of the paper is organized as follows: Section II presents the proposed approach. Experimental results are presented and analyzed in Section III. Finally, we conclude and discuss future research directions in Section IV.

2 Proposed System

The image is defined as the set of combination of color information, texture and shape of the object in the image. Among these features, color and texture features are considered in this paper. The first and foremost step is to represent the images in terms of features. The visual signature of the i^{th} image is made up of different feature vectors, composed by: M_{ch} color histogram bins c_i^{ch}, M_{cm} color moments c_i^{cm} [3], M_{edh} edge direction histogram c_i^{edh} [4] and M_{wt} wavelet texture feature values c_i^{wt} [7]. The feature vector $c_i = [c_i^{ch} + c_i^{cm} + c_i^{edh} + c_i^{wt}]$ of dimension D= $M_{ch} + M_{cm} + M_{edh} + M_{wt}$ provides the overall description of the image. The feature vectors of query image are computed online and the feature vectors of stored database images are computed offline. From there, each image is represented as feature vector in *D*-dimensional space. After the mapping of query image feature vector and stored database image feature vector, the system shows the most M_{FB} nearest image from the entire database, based on weighted Euclidean distance between feature vector pairs to the user for collecting the first feedback. Mathematically expressed as

$$Dist(c_q ; c_S) = WMSE(c_q^{ch} ; c_s^{ch}) + WMSE(c_q^{cm} ; c_s^{cm}) + WMSE(c_q^{edh} ; c_s^{edh}) + WMSE(c_q^{wt} ; c_s^{wt}) \quad (1)$$

where c_q is the query feature vectors and c_s is the stored database feature vectors; $S=1,\ldots,M_{DB,}$, where M_{DB} is the total number of database images. *WMSE* is the weighted Euclidean distance calculated between a pair of feature vectors:

$$WMSE(c_q ; c_s) = \frac{1}{N} \sum_{j=1}^{N} (c_{qj} - c_{sj})^2 w_j^k \tag{2}$$

where w_j^k is a vector of weights associated to the features at k^{th} iteration and N is equal to M_{ch} or M_{cm} or M_{edh} or M_{wt}. At first iteration, all the features are equally important i.e., $w_j^k = 1; j = 1,.....,N;$ Then, the user tags the images as relevant and irrelevant according to their mental view of query and it is updated during all the iterations. Now, the feature re-weighting algorithm is used which is based on a set of statistical characteristics [4]. Based on the concept of *dominant range* and *confusion set*, it is feasible to calculate the *discriminant ratio* δ_f^k on the f^{th} feature ($f = 1,2,.....,D$) at the k^{th} iteration which shows the ability of this feature to separate irrelevant images from relevant ones. The updated weight is then computed as follows

$$w_f^{k+1} = \frac{\delta_f^k}{\sigma_f^{k,R}} \tag{3}$$

Where $\sigma_f^{k,R}$ is the standard deviation of f^{th} feature of the relevant image subset at the k^{th} iteration which is modified [8] with normalization factor, thereby it limits the maximum weight to 1. The weight updating is to emphasize the most significant one.

Now, the retrieval problem is modeled as an optimization process by the use of firefly algorithm. It is worth noting that it is possible to view the swarm of agents A_n or swarm of fireflies as query points that will explore the D-dimensional search space, which is made up of image features ($f = 1,2,.....,D$)with its light intensity.

The decision variables of firefly algorithm are the four feature vectors as M_{ch} , M_{cm}, M_{edh} and M_{wt}. The brightness of light intensity is associated with the objective function which is related to the sum of weighted Euclidean distance between the query image and the stored database image in D-dimensional search space. The lower the weighted Euclidean distance between the query image and the stored database image feature vector, higher the light intensity and thus more the attractiveness. There are two phases of firefly algorithm which are described as follows [5]:

In the first phases, given that there exists an n number of swarm of fireflies with $M_{FB} \leq n < M_{DB}$ in which each firefly is determined by the light intensity and x_i represents a solution for an agent i, whereas $f(x_i)$ indicates its corresponding fitness value. Here the brightness of a swarm of agent I is equivalent to the fitness value.

$$I_i = f(x_i) \quad 1 \leq i \leq n \tag{4}$$

In the second phase, the attractiveness β of the firefly is proportional to the light intensity received by the adjacent fireflies. Suppose β_0 is the attractiveness with distance $r = 0$, so for two fireflies i and j at locations x_i and x_j, their attractiveness is calculated as

$$\beta_r(i, j) = \beta_0 e^{(-\gamma r(i,j)^2)} \tag{5}$$

$$r(i, j) = | x_i - x_j | \tag{6}$$

where *r(i, j)* denotes the distance between fireflies *i* and *j*, γ denotes the light absorption coefficient. Suppose firefly *j* is brighter than firefly *I* of the input image, then firefly *i* will move to a new location as

$$x_i(t+1) = x_i(t) + \beta_0 e^{(-\gamma r^2)}(x_j - x_i) + \alpha\varepsilon_i \tag{7}$$

where the first term is the current position of a firefly, the second term is due to attractiveness and the third term is the randomization with the vector of random variable ε_i being drawn from the normal distribution [5] and $\alpha \in [0,1]$.

When the value of *r* (distance between two fireflies) is small/large, the firefly will move a large/small distance which will affect the computation time of this algorithm. Also the agents move in a fixed step length, so the firefly algorithm may loose of searching local feature space and traps into several local optimum solutions. Thus a random step length is used for firefly movement where the firefly initially search the feature space globally and in the end of iterative process the firefly exploit feature space locally to obtain better solutions and also adaptively it changes the step length by the time which overcomes the above drawback. The weight of consecutive random step length for α is determined by the following equation whose value is always less than one and it depends on maximum iteration number $iter_{max}$ and present iteration number *iter*[9].

$$W_{iter} = A + \frac{(iter_{max} - iter)^n}{(iter_{max})^n} + (A - B) \tag{8}$$

where A=0 & B=1since $\alpha \in [0,1]$. W_{iter} is between A & B and its value reduces by the time. m would be a linear or non-linear co-efficient and it depends on the dimension of each agent. If the dimension is high, the value of m is low which means the algorithm can converge more accurate. Its value is determined by

$$m = 10^{(-dimension)} \tag{9}$$

If the firefly step length follows Gaussian distribution, then the random walk movement becomes the Brownian motion [5]. At the end of each iteration in firefly algorithm, normal Gaussian distribution is introduced in order to move all of the fireflies to global best and is shown in the following equation.

$$p = f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\Pi\sigma^2}} e^{\left(\frac{-|x-\mu|^2}{2\sigma^2}\right)} \tag{10}$$

where μ and σ^2 are its mean, variance and *x* is the error between the fitness value and best solution of firefly *i*.

$$x = f(b_{best}) - f(x_i) \tag{11}$$

Because of the use of standard normal distribution, $\mu=0$ and $\sigma^2=1$. Then from this Gaussian distribution, a random number is drawn which is related to swarm of each agent probability (*p*). The behavior of agent is then introduced by [6]:

$$x_i(t+1) = x_i(t) + \alpha * (1 - p) * U(x, z) \tag{12}$$

where $U(x, z)$ is a random number between 0 & 1. If the evaluated new solution cost $x_i(t+1)$ better than the current position $x_i(t)$, then the firefly will move towards that new position. The updating process is done consequently at further iteration. The value of objective function, attractiveness and movement of firefly towards other firefly are recalculated according to (4), (5) and (7) respectively when new relevant images are chosen by user. Thereby the firefly moves towards the new area in the feature space where the new relevant images may be found. After user feedback, the process of feature reweighting and firefly updating is iterated until result of search satisfies user.

The most important point in an optimization process is to define the target function that is to be minimized or maximized which is said to be fitness. Taking into account of irrelevant and relevant images, the weight cost function defined [8] by equation (13) expresses the fitness associated to the solution space found by the swarm of firefly A_n.

$$\phi^k(A_n) = \frac{1}{N^k_{rel}} \sum_{r=1}^{N_{rel}} Dist(A^k_n ; x^k_r) + \frac{1}{\frac{1}{N^k_{irr}} \sum_{r=1}^{N_{irr}} Dist(A^k_n ; x^k_i)} \tag{13}$$

Where x^k_i ; $i = 1,..., N^k_{irr}$ and x^k_r ; $r = 1,..., N^k_{rel}$ are the images in the irrelevant and relevant image subsets, respectively. As the distance between the firefly A_n and irrelevant images grows, the fitness associated to the solution space depends only on relevant images.

3 Experimental Results

In our experiments, we used the Corel database [10] covering a wide range of semantic categories from natural scenes to artificial objects. The dataset is partitioned

Table 1. Average precision values (in percent) of best found results

Category	GA+RF in CBIR	PSO+RF in CBIR	Standard FA+RF in CBIR	GD-FA+RF in CBIR
Butterfly	78.27	79.10	90.83	93.322
Buildings	27.66	78.49	95.39	96.966
Hills	78.57	75.60	93.61	95.572
Flowers	78.97	78.75	91.63	93.366
Earth	78.87	76.75	99.73	99.431
Sky	78.37	59.54	94.27	95.572
Tree	78.57	79.11	95.92	96.322
Boat	27.96	78.79	92.26	89.774
Bird	78.07	79.11	98.95	99.431
Statue	78.17	67.2	96.57	96.966

into 10 categories and each category is represented by 250 images, for a total of 2500 images. All the experiments were implemented in Matlab, running on a personal computer with Intel Dual Core 3GHZ processor and 4 GB RAM. From Table 1, it is inferred that the proposed method outperforms other methods like PSO and GA. Also due to the inclusion of Gaussian distribution in firefly algorithm (GD-FA), the accuracy is increased except for the case of "boat" category.

4 Conclusion and Future Work

This paper has presented the Firefly based content based image retrieval optimization. The CBIR system has been implemented with the relevance feedback mechanism and for the optimization the objective function for the firefly has been designed with image color parameters and texture parameters. According to the performance with respect to the image precision, firefly has a deeper precision accuracy than PSO and GA method optimization. In addition, by the use of Gaussian distribution all the agents moved to global best solutions and got rid of several local optimum solutions. Hence, it is highly efficient, robust and highly rapid for image accuracy based application. The future research work focuses on improving the retrieval quality by using region based image retrieval.

References

1. Smeulders, A.W., ...Jain, R.: Content- based image retrieval at the end of the early years. IEEE Trans. Pattern Anal. Mach. Intell. 22(12), 1349–1380 (2000)
2. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Comput. Surv. 40(2) (2008)
3. Grigorova, A., ...Huang, T.S.: Content based image retrieval by feature adaptation and relevance feedback. IEEE Trans. Multimedia 9(6), 1183–1192 (2007)
4. Wu, Y., Zhang, A.: A feature re-weighing approach for relevance feedback in image retrieval. In: Proc. IEEE Int. Conf. Image Processing (ICIP 2002), vol. 2, pp. 581–584 (2002)
5. Yang, X.-S.: Firefly algorithms for multimodal optimization. In: Watanabe, O., Zeugmann, T. (eds.) SAGA 2009. LNCS, vol. 5792, pp. 169–178. Springer, Heidelberg (2009)
6. Farahani, S.M., ...Meybodi, M.R.: A Gaussian Firefly Algorithm. International Journal of Machine Learning and Computing 1(5) (December 2011)
7. Deselaers, T., Keysers, D., Ney, H.: Features for image retrieval: An experimental comparison. Inf. Retriev. 11(2), 77–107 (2008)
8. Broilo, M., De Natale, F.G.B.: A Stochastic Approach to Image Retrieval Using Relevance Feedback and Particle Swarm Optimization. IEEE Trans. Multimedia 12(4) (June 2010)
9. Yazdani, D., Meybodi, M.R.: AFSA-LA: A New Model for Optimization. In: Proceedings of the 15th Annual CSI Computer Conference (CSICC 2010), February 20-22 (2010)
10. The corel database for content based image retrieval, https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval

An Integrated Method for Functional Analysis of Microbial Communities by Gene Ontology Based on 16S miRNA Gene

Suping Deng and Kai Yang

School of Electronics and Information Engineering, Tongji University,
Caoan Road 4800, Shanghai, 201804, P.R. China
dsptk2003@126.com, yangzhenhaoyu@163.com

Abstract. The environment and human serve as elaborate hosts of microbes, including a diversity of commensal and pathogenic bacteria that contribute to both health and diseases. 16S rRNA genes are useful for community profiling, or determining the abundance of each kind of microbe. The purpose of our study is to analyze the similarity among microbial communities on functional state after assigning 16S rRNA sequences from all microbial communities to species. It's an important addition to the species-level relationship between two compared communities, and can quantify their differences in function. To accomplish this aim, we downloaded all functional annotation data of microbiota from related datasets. It's developed to identify the functional distribution and the significantly enriched functional categories of microbial communities. Exploration of the function relationship between two sets of species assemblages will be a key result of microbiome studies and may provide new insights into assembly of a wide range of ecosystems.

Keywords: Microbial community, 16s rRNA, Gene Ontology enrichment component, GO-terms Semantic Similarity.

1 Introduction

Specialized microbial communities inhabit the environment, skin, mucosal surfaces, and gastrointestinal tract of humans from birth until death [1, 2]. Human beings rely on their native microbiota for nutrition and resistance to colonization by pathogens [3–6]; furthermore, recent discoveries have shown that symbiotic microbes make essential contributions to the development, metabolism, and immune response of the host [7–10]. Co-evolved, beneficial, human–microbe interactions can be altered by many aspects of a modern lifestyle, including urbanization, global travel, and dietary changes [1]. Because many chemical transformations in human vivo are mediated by specific microbial populations [11], with implications for cancer [12,13]and obesity[14,15],under other conditions[16], changes in the composition of the human microbiota could have important but undiscovered health effects.

Over the past decade, cultivation independent molecular techniques, particularly those based on small subunit ribosomal RNA(16S rRNA)gene, have given us a

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 219–224, 2013.

broader and less biased view of diversity and abundance of the microbiota[17,18]. Full-length 16S rRNA sequences offer the highest possible degree of taxonomic resolution using this gene, but the cost of Sanger sequencing limits our ability to survey the less-abundant members of this diverse community. But currently, microbial community profiling using 16S rRNA is undergoing a renaissance[19] as high-throughput techniques such as barcoded pyrosequencing allow us to gain deep view into hundreds of microbial communities simultaneously and have less cost than Sanger sequencing to allow more samples to be processed.

There are still some questions not being resolved. For example, one recent intriguing result is that species-level variability appears to be associated with extensive functional redundancy, in which completely different microbial communities converge on the same functional state [20]. So the purpose of our work is to analyze the similarity between two biological communities on the functional state. It's an important addition to the species-level relationship between two compared microbial communities, and it can quantify their differences in function after using 16S rRNA to assign sequences of communities to taxonomy. 16S rRNA can be used for community profiling, or determining the abundance of each kind of microbe [21]. Several tools can be used for community profiling, such as the Ribosomal Database Project (RDP) naive Bayesian classifier, DOTUR, Unifrac. Then, our purpose is to use taxonomy ids identified by RDP naive Bayesian classifier to do the Gene ontology (GO) annotation, and do the GO enrichment analysis and the similarity of each set of GO terms of two biological communities. Next we will cluster all communities according to the similarity of two communities.

Exploration of this function relationship between two sets of species assemblages may provide new insights into microbiome studies. And it provides an effective clue to define biological health, distinguish between disease susceptibility.

2 Methods and Results

2.1 Functional Annotation Dataset of Microbiota

The raw data of our study was NCBI taxonomy ids identified using RDP. Ribosomal Database Project (RDP) naive Bayesian classifier assigned 16S rRNA sequences of all microbial communities to taxonomy.

The purpose of this step is using the NCBI taxonomy ids (raw data) to find and download corresponding gene functional annotation data of all annotated microbial species. Currently, some FTPs of website have functional annotation dataset of microbiota, such as ftp: // ftp. ncbi. nih. gov / gene / DATA / GENE_INFO/(NCBI), ftp: // ftp. geneontology. org / pub /go / (GOA), ftp://ftp. expasy. org/databases/ hamap/ complete_proteomes /entries/(HAMAP).

We compared the three FTPs of website to find which one contains more microbial species annotation data. By comparing the results, we found the HAMAP website FTP (ftp://ftp.expasy.org) have more annotations. Complete microbial proteomes compiled in the framework of the HAMAP project (High quality automated and manual annotation of microbial proteomes).

Then according to that, we downloaded 1065 functional annotation data files about microbiota from HAMAP, and extracted useful information of them to a text file with PERL. The usage of the text file is that, when we search keyword about NCBI_TaxID or GeneID, it could export corresponding functional annotation by a PERL program.

Gene Ontology enrichment analysis and semantic similarity measures of microbial community.

According to the above step, corresponding GO terms could be gained by using NCBI_TaxID or GeneID of microbial species in each microbial community. In order to fully understand the relationship between the two compared microbial communities on functional level, GO-terms enrichment analysis and semantic similarity measures were done.

Gene Ontology enrichment analysis.

Gene ontology (GO) includes terms of three levels: biological processes (BP), cellular components (CC) and molecular function (MF).

In this section, we researched the enrichment of GO terms of each microbial community at biological processes (BP) level. The enrichment analysis was run with the topGO package[22]. Fisher's exact test is based on gene counts. So, in this study, the classic Fisher method was chosen.

In order to explain this GO enrichment analysis, an example was given as followed. In the example, four microbial community samples were set artificially. The following microbial were selected randomly in each sample:

Semantic Similarity Measures between sets of GO-terms of Each Microbial Community.

(1)Sample1 contained Prochlorococcus marinus (strain NATL1A) [NCBI_TaxID =167555], Pseudomonas fluorescens (strain Pf0-1) [NCBI_TaxID=205922], Exiguobacterium sibiricum (strain DSM 17290/JCM 13490/255-15) [NCBI_TaxID =262543];

(2)Sample2 contained Ehrlichia chaffeensis (strain Arkansas) [NCBI_TaxID =205920], Haemophilus somnus (strain 2336) [NCBI_TaxID=228400], Salmonella newport (strain SL254) [NCBI_TaxID=423368];

(3)Sample3 contained Acaryochloris marina (strain MBIC 11017) [NCBI_TaxID=329726], Shewanella baltica (strain OS195) [NCBI_TaxID=399599], Aggregatibacter aphrophilus (strain NJ8700) [NCBI_TaxID=634176];

(4)Sample4 contained Cryptosporidium hominis [NCBI_TaxID=237895], Acinetobacter baumannii (strain SDF) [NCBI_TaxID =509170]

Six GO_BP terms with $p<0.01$ were significantly enriched in sample1 (Table 1), seven GO_BP terms in sample2 (Table 2), ten GO_BP terms in sample3 (Table 3), and four GO_BP terms in sample4 (Table 4). It is shown that Sample4 involves most types of biological process among them under the condition that p is less than 0.01.

Four methods have been presented to determine the semantic similarity of two GO terms based on the annotation statistics of their common ancestor terms (Resnik[23], Jiang[24], Lin[25] and Schlicker[26]). Wang [27] proposed a new method to measure the similarity based on the graph structure of GO.

In this study, the GO-terms semantic similarity analysis was running with the GOSemSim package using Wang method. The GO-terms of each microbial

Table 1. Gene ontology enriched in sample1 (p<0.01)

GO.ID	Terms	GO level	Classic Fisher
GO:0042891	antibiotic transport	BP	0.0055
GO:0015758	glucose transport	BP	0.0094
GO:0019478	D-amino acid catabolic process	BP	0.0094
GO:0046416	D-amino acid metabolic process	BP	0.0094
GO:0015893	drug transport	BP	0.010
GO:0042493	response to drug	BP	0.010

Table 2. Gene ontology enriched in sample2 (p<0.01)

GO.ID	Terms	GO level	Classic Fisher
GO:0006529	asparagine biosynthetic process	BP	0.00097
GO:0019538	protein metabolic process	BP	0.00186
GO:0044267	cellular protein metabolic process	BP	0.00458
GO:0006470	protein amino acid dephosphorylation	BP	0.00830
GO:0006528	asparagine metabolic process	BP	0.00830
GO:0046689	response to mercury ion	BP	0.00830
GO:0006421	asparaginyl-tRNA aminoacylation	BP	0.00980

Table 3. Gene ontology enriched in sample3 (p<0.01)

GO.ID	Terms	GO level	Classic Fisher
GO:0006570	tyrosine metabolic process	BP	0.0034
GO:0000270	peptidoglycan metabolic process	BP	0.0039
GO:0030203	glycosaminoglycan metabolic process	BP	0.0039
GO:0006022	aminoglycan metabolic process	BP	0.0052
GO:0009372	quorum sensing	BP	0.0095
GO:0048872	homeostasis of number of cells	BP	0.0095
GO:0048874	homeostasis of number of cells in a free...	BP	0.0095
GO:0006023	aminoglycan biosynthetic process	BP	0.010
GO:0006024	glycosaminoglycan biosynthetic process	BP	0.010
GO:0009252	peptidoglycan biosynthetic process	BP	0.010

Table 4. Gene ontology enriched in sample4 (p<0.01)

GO.ID	Terms	GO level	Classic Fisher
GO:0008610	lipid biosynthetic process	BP	0.002
GO:0006633	fatty acid biosynthetic process	BP	0.004
GO:0044255	cellular lipid metabolic process	BP	0.008
GO:0006415	translational termination	BP	0.01

community was shown in Table 5. Then, cluster dendrogram of the four samples could draw by using the output value of GOSemSim(Fig. 1). In Table 5, it is shown that Sample4 is most similar to Sample1, but most different from Sample3. The conclusion can also be drawn from Fig. 1.

Table 5. Semantic Similarity of sets of GO-terms of Microbial Community

	Sample1	Sample2	Sample3	Sample4
Sample1		0.489	0.343	0.288
Sample2			0.426	0.475
Sample3				0.324
Sample4				

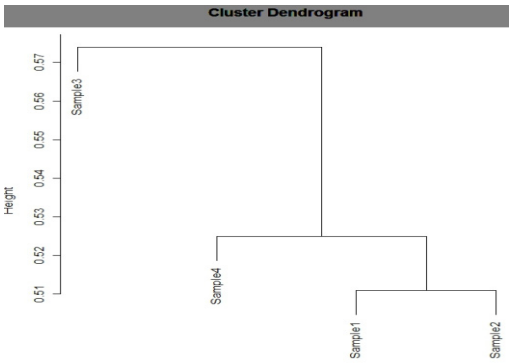


Fig. 1. Cluster dendrogram of the four samples: the value of height was equal to 1 minus output value of GOSemSim

3 Conclusion

In the paper, we used Ribosomal Database Project (RDP) naive Bayesian classifier to assign 16S rRNA sequences of all microbial communities to taxonomy. Then we analyzed the similarity between two microbial communities on the functional state.

Through the experimental results, we could conclude that the functional annotation of microbial community provides an effective clue to define biological health, distinguish between disease susceptibility, and diagnose disease. And it brings inexhaustible power to drug development, vaccine development, new diagnostic markers and targeted therapies.

Acknowledgment. This work was supported by the grants of the National Science Foundation of China, No. 61133010.

References

1. Dethlefsen, L., McFall-Ngai, M., Relman, D.A.: An Ecological and Evolutionary Perspective on Human-microbe Mutualism and Disease. Nature 449, 811–818 (2007)
2. Ley, R.E., Peterson, D.A., Gordon, J.I.: Ecological and Evolutionary Forces Shaping Microbial Diversity in the Human Intestine. Cell 124, 837–848 (2006)
3. Guarner, F.: Enteric Flora in Health and Disease. Digestion 73, 5–12 (2006)
4. Brook, I.: The Role of Bacterial Interference in Otitis, Sinusitis and Tonsillitis. Otolaryngol Head Neck Surg 133, 139–146 (2005)
5. Servin, A.L.: Antagonistic Activities of Lactobacilli and Bifidobacteria Against Microbial Pathogens. FEMS Microbiol Rev. 28, 405–440 (2004)
6. Reid, G., Bruce, A.W.: Probiotics to Prevent Urinary Tract Infections: the Rationale and Evidence. World J. Urol. 24, 28–32 (2006)
7. Li, M., Wang, B., Zhang, M., et al.: Symbiotic Gut Microbes Modulate Human Metabolic Phenotypes. Proc. Natl. Acad. Sci. 105, 2117–2122 (2008)

8. Hooper, L.V.: Bacterial Contributions to Mammalian Gut Development. Trends Microbiol. 12, 129–134 (2004)
9. Mazmanian, S.K., Liu, C.H., Tzianabos, A.O., Kaspe, R.D.L.: An Immunomo-dulatory Molecule of Symbiotic Bacteria Directs Maturation of the Host Immune System. Cell 122, 107–118 (2005)
10. Rakoff-Nahoum, S., Paglino, J., Eslami-Varzaneh, F., et al.: Recognition of Commensal Microflora by Toll-like Receptors is Required for Intestinal Homeostasis. Cell 118, 229–241 (2004)
11. Nicholson, J.K., Holmes, E., Wilson, I.D.: Gut Microorganisms, Mammalian Metabolism and Personalized Health Care. Nat. Rev. Microbiol. 3, 431–438 (2005)
12. O'Keefe, S.J., Chung, D., Mahmoud, N., et al.: Why Do African Americans Get More Colon Cancer than Native Africans. J. Nutr. 137,175S–182S (2007)
13. McGarr, S.E., Ridlon, J.M., Hylemon, P.B.: Diet, Anaerobic Bacterial Metabolism, and Colon Cancer: A Review of the Literature. J. Clin. Gastro. enterol. 39, 98–109 (2005)
14. Ley, R.E., Backhed, F., Turnbaugh, P., et al.: Obesity Alters Gut Microbial Ecology. Proc. Natl. Acad. Sci. 102, 11070–11075 (2005)
15. Backhed, F., Ding, H., Wang, T., et al.: The Gut Microbiota as An Environmental Factor That Regulates Fat Storage. Proc. Natl. Acad. Sci. 101, 15718–15723 (2004)
16. Stewart, C.S., Duncan, S.H., Cave, D.R.: Oxalobacter Formigenes and Its Role in Oxalate Metabolism in The Human Gut. FEMS Microbiol. Lett. 230, 1–7 (2004)
17. Zoetendal, E.G., Collier, C.T., Koike, S., et al.: Molecular Ecological Analysis of The Gastrointestinal Microbiota:A Review. J. Nutr. 134, 465–472 (2004)
18. Mai, V., Morris, J.G.: Colonic Bacterial Flora:Changing Understandings in The Molecular Age. J. Nutr. 134, 459–464 (2004)
19. Tringe, S.G., Hugenholtz, P.: A Renaissance for The Pioneering 16S rRNA Gene. Curr. Opin. Microbiol. 11, 442–446 (2008)
20. Dinsdale, E.A., Edwards, R.A., Hall, D., et al.: Functional Metagenomic Profiling of Nine Biomes. Nature 452, 629–632 (2008)
21. Pace, N.R.: A Molecular View of Microbial Diversity and The Biosphere. Science 276, 734–740 (1997)
22. Alexa, A., Rahnenfuhrer, J., Lengauer, T.: Improved Scoring of Functional Groups from Gene Expression Data by Decorrelating GO Graph Structure. Bioinformatics 22, 1600–1607 (2006)
23. Resnik, P.: Semantic Similarity in A Taxonomy: An Information-Based Measure and Its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research 11, 95–130 (1999)
24. Jay, J., Jiang, D., Conrath, W.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: Proceedings of 10th International Conference on Researc In Computational Linguistics (1997)
25. Dekang, L.: An Information-Theoretic Definition of Similarity. In: Proceedings of the 15th International Conference on Machine Learning, pp. 296–304 (1998)
26. Schlicker, A., Domingues, F.S., Aijhrer, J.R., et al.: A New Measure for Functional Similarity of Gene Products Based on Gene Ontology. BMC Bioinformatics 7, 302 (2006)
27. Wang, J.Z., Du, Z., Payattakool, R., et al.: A New Method to Measure The Semantic Similarity of GO terms. Bioinformatics 23, 1274–1281 (2007)

Possible miRNA Coregulation of Target Genes in Brain Regions by Both Differential miRNA Expression and miRNA-Targeting-Specific Promoter Methylation

Y.-H. Taguchi

Department of Physics, Chuo University, Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan
tag@granular.com

Abstract. We computationally reanalyzed public domain data set deposited to gene expression omnibus, of mRNA expression, miRNA expression and promoter methylation pattern in four brain regions, i.e., frontal cortex, temporal cortex, pons and cerebellum. Then we found that more than hundreds of both miRNA regulation of target genes and miRNA-targeting-specific promoter methylation are significant on all six pairwise comparisons among the above mentioned four brain regions. We also showed that some of miRNA regulation of target genes is highly correlated with both or either of miRNA-targeting-specific promoter methylation and differential miRNA expression. We concluded that the combinatorial analysis of miRNA regulation of target genes, miRNA-targeting-specific promoter methylation and differential miRNA expression can figure out brain region specific contribution of miRNAsto brain functions and developments.

Keywords: miRNA, promoter methylation, brain regions.

1 Introduction

miRNAs are short non-coding RNAs that are believed to suppress target gene expression through complementary sequence matching between "seed" region of miRNA and 3' untranslated region (UTR) of target genes[1]. Since biogenesis and functionality of miRNAs were relatively well-known compared with other non-coding RNAs, there were huge number of papers published about miRNAs. miRNAs are generally supposed to regulate cellular processes related to animal development[2], differentiation and several diseases/tumor formation. Thus, miRNAs are often regarded to be candidates of tumor suppressor [3] or cancer biomarkers [4]. miRNAs were also used for the reprogramming [5]. As such, miRNAs are considered to play critical roles over the wide range of biological processes.

Recently, miRNA expression in brain attracts the interest of many researchers [6–9]. Although there are extensive researches about miRNA regulation of target genes [6, 7], it is generally believed that majorities of miRNA regulation of genes are indirect [10] and not all target genes are directly regulated by miRNAs. In this sense, in order to understand miRNA regulation of gene expression in brain regions, we also

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 225–230, 2013.

need to know other mechanisms that regulate miRNA target genes than miRNAs, in brain regions.

Promoter methylation is generally thought to suppress gene expression [11]. Suppression of gene expression by promoter methylation is often important. For example, aberrant promoter methylation is often related to cancers [12, 13]. Promoter methylation also plays critical roles in reprogramming [14]. In spite of the importance of promoter methylation, correlation between promoter methylation and miRNA regulation of target genes was rarely discussed. One of seldom researches about the coregulation between promoter methylation and miRNA regulation of target genes was recently conducted by Su et al [15]. They found that promoters of genes not targeted by miRNAs have tendencies to be methylated. Although there were no follow-up studies of it, we recently found that miRNA-targeting-specific promoter methylation takes place over many cell lines [16, 17]. In this paper, we report that miRNA-targeting-specific promoter methylation also exists between distinct brain-regions in a brain-region specific manner.

Moreover, some miRNA regulation of target genes turned out to be controlled by not only differential miRNA expression itself but also miRNA-targeting-specific promoter methylation.

2 Mutual Relationships between miRNA Regulation of Genes, miRNA-Targeting-Specific Promoter Methylation and Differential miRNA Expression

We investigated miRNA regulation of target genes and miRNA-targeting-specific promoter methylation among frontal cortex (FCTX), temporal cortex (TCTX), pons (PONS), and cerebellum (CRBLM), based on the P-values, $P^{ll'}_{mj,<}$ s or $P^{ll'}_{mj,>}$ s, that estimate miRNA regulation of target genes and miRNA-targeting-specific promoter methylation, for the mth miRNA of jth sample between lth and l'th brain regions. < (>) stands for l'th regions have more upregulated (downregulated) genes or hypermethylated (hypomethylated) promoters than lth region. Fig. 1 illustrates the results of this analysis. It is clear that target genes of substantial number of miRNAs are up/downregulated between these four brain regions. It is also sure that substantial number of miRNAs' target genes' promoters are hyper/hypomethylated between these four brain regions. This strongly suggests that both miRNA regulation of target genes and miRNA-targeting-specific promoter methylation play critical roles to make these four brain regions develop and function differently from each other. Then, it is important to understand how these two cooperatively regulate target genes. In order to understand the mutual relationship between miRNA regulation of target genes and miRNA-targeting-specific promoter methylation, we computed the correlation coeffcient of mean rank of P-values, $\rho^{miRNA,Methyl}_{ll'}$, for six pairwise comparisons among frontal cortex, temporal cortex, pons and cerebellum (see Fig. 1). Here the means were taken over all of samples in each brain region.

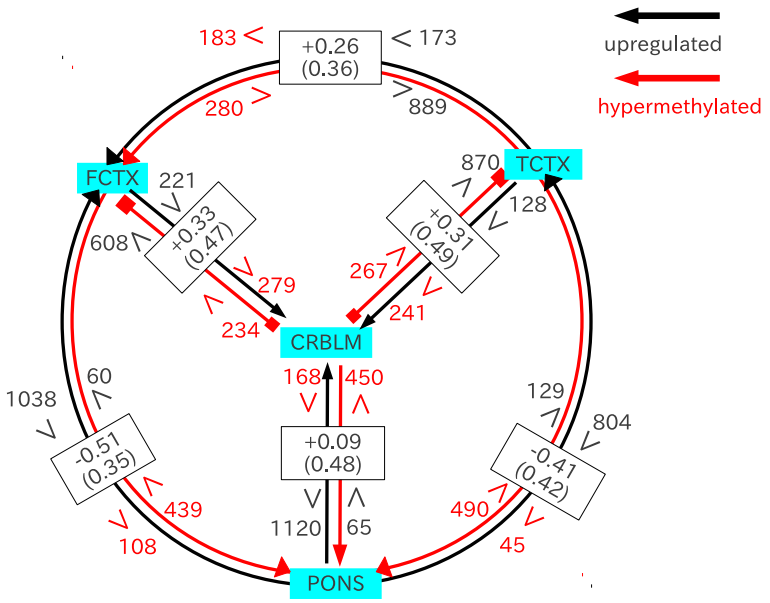


Fig. 1. Schematic illustration of the relationship between miRNA regulation of target genes and miRNA-targeting-specific promoter methylation. Arrows/segments indicate up/downregulation of miRNA target genes and miRNA-targeting-specific promoter methylation. Black (red) numbers next to inequality signs are the average number of miRNAs whose target genes are significantly up/downregulated (whose target genes promoters are hyper/hypomethylated). There are 280 (183) miRNAs whose target genes promoters are significantly hyper(hypo)methylated in FCFX compared with TCTX. Similarly, there are 889 (173) miRNAs whose target genes are significantly up(down)regulated in FCFX compared with TCTX. Since 280 is large enough than 183, promoters of genes in FCFX is regarded to be hypermethylated than TCTX in the miRNA-centric-view, thus red arrow directs from TCTX to FCTX. Similarly, since 889 is large enough than 173, genes in FCFX is regarded to be upregulated than TCTX in the miRNA-centric-view, thus black arrow directs from TCTX to FCTX. The numbers in rectangular indicate Spearman correlation coeffcients between miRNA regulation of target genes and miRNA-targeting-specific promoter methylation, $\rho_{ll'}^{miRNA,Methyl}$. Those in parentheses are the standard deviations of Spearman correlation coeffcients, $\Delta\rho_{ll'}^{miRNA,Methyl}$.

Excluding a pair of cerebellum and pons, correlation coeffcients for other five pairwise comparisons take values ranging from 0.25 to 0.51. These values can be regarded to be large enough if we consider that the number of *P*-values attributed to each brain region is as large as *M* that is equal to the number of miRNAs. Actually, the *P*-values attributed to each correlation coeffcient are less than 2×10^{-16}. This means, the correlation between miRNA regulation of target genes and miRNA-targeting-specific promoter methylation is highly significant independent of pairs of

brain regions. The remaining and the smallest correlation coeffcient is that attributed to the pair of cerebellum and pons.

However, its value is still as large as 0.09 and the attributed P-values to it is as small as 4×10^{-5}, thus is highly significant too, although the correlation itself cannot be said to be large enough. In order to confirm the firm correlation between miRNA regulation of target genes and miRNA-targeting-specific promoter methylation, the root mean squared averages of correlation coffcients of each sample, $\Delta\rho_{ll'}^{miRNA,Methyl}$, were also computed. Excluding the pair of frontal cortex and pons where the absolute value of $\rho_{ll'}^{miRNA,Methyl}$ was the maximum, $\Delta\rho_{ll'}^{miRNA,Methyl}$ is larger than the absolute value of $\rho_{ll'}^{miRNA,Methyl}$. This means, correlation coffcients within each sample are not small but only averaged value over samples is small because of the appearance of both positive and negative correlation with equal probabilities in each sample. As a result, miRNA regulation of target genes and miRNA-targeting-specific promoter methylation are significantly correlated with each other. One may notice that the signs of correlation coeffcients, $\rho_{ll'}^{miRNA,Methyl}$, are neither positive definite nor negative definite. One may think that they should be positive definite because both promoter methylation and miRNA targeting should suppress gene expression.

However, since genes targeted by miRNAs are expected to be downregulated (upregulated) only when miRNA itself is upregulated (downregulated), there are no reasons to expect that the correlation coeffcients between miRNA regulation of target genes and miRNA-targeting-specific promoter methylation always takes positive or negative values.

Next, we have introduced the multivariate regression models between miRNA regulation of target genes, miRNA-targeting-specific promoter methylation and differential miRNA expression, together with gender and age information (methodological details are not shown). In contrast to the above discrepancy, we have found some significant correlations between those variables. It was found that not all of features, i.e., miRNA regulation of target genes, miRNA-targeting-specific promoter methylation and differential miRNA expression together with age and gender, were always correlated but were selectively correlated. That is, dependent upon the miRNA considered, a combination of limited part of these features is correlated. In order to quantize these selective correlations, we picked up the combination of features significantly correlated with each other for each miRNA (methodological details are not shown). Table 1 lists the miRNAs selected for each pair of brain regions based on the criterion described in the above, i.e., miRNAs whose differential expression is significantly correlated to miRNA regulation of target genes. To our knowledge, for the first time, we showed that miRNA regulation of target genes are mediated by both differential miRNA expression and miRNA-targeting-specific promoter methylation.

Acknowledgment. This research was supported by KAKENHI 23300357 and A-Step; Adaptable and Seamless Technology Transfer Program through target-driven R&D, Exploratory Research, AS242Z00112Q.

Table 1. miRNAs supposed to regulate target genes for six pairwise comparisons among four brain regions, frontal cortex (FCTX), temporal cortex (TCTX), pons (PONS), and cerebellum (CRBLM). "Reciprocal" ("nonreciprocal") indicates relationship between miRNA expression and target gene mRNA is reciprocal (nonreciprocal). miRNAs in bold face appear more than once. Underlined miRNAs were previously reported to be related to brain develoment/diseases [18–20].

CRBLM vs FCTX		CRBLM vs PONS		CRBLM vs TCTX	
reciprocal	nonreciprocal	reciprocal	nonreciprocal	reciprocal	nonreciprocal
hsa-miR-181c-5p	hsa-miR-200a-5p	hsa-miR-20a-5p	hsa-let-7b-5p	hsa-miR-210	hsa-miR-99a-5p
hsa-miR-135a-5p	hsa-miR-381	hsa-miR-23a-3p	hsa-let-7e-5p		hsa-miR-191-5p
hsa-miR-137	**hsa-miR-202-3p**	hsa-miR-148a-3p	hsa-miR-197-3p		**hsa-miR-99b-5p**
hsa-miR-363-3p	hsa-miR-561-3p	hsa-miR-10a-5p	hsa-miR-181b-5p		hsa-miR-617
hsa-miR-369-3p	hsa-miR-568	hsa-miR-221-3p	hsa-let-7i-5p		
hsa-miR-487a	hsa-miR-618	hsa-miR-223-3p	hsa-miR-9-5p	FCFX vs PONS	
hsa-miR-514a-3p	**hsa-miR-630**	hsa-miR-1	hsa-miR-126-3p	hsa-miR-365a-3p	hsa-miR-302d-3p
hsa-miR-553		hsa-miR-133a	hsa-miR-134	hsa-miR-378a-5p	hsa-miR-432-5p
hsa-miR-554		**hsa-miR-137**	hsa-miR-154-3p		hsa-miR-595
hsa-miR-655		hsa-miR-146a-5p	hsa-miR-299-5p		
hsa-miR-421		hsa-miR-452-5p	**hsa-miR-99b-5p**	FCTX vs TCTX	
		hsa-miR-484	hsa-miR-377-3p	has-miR-373-3p	hsa-miR-24-3p
		hsa-miR-511	hsa-miR-383		hsa-miR-485-5p
		hsa-miR-515-5p	hsa-miR-431-5p		hsa-miR-766-3p
		hsa-miR-571	hsa-miR-329		
		hsa-miR-549	hsa-miR-485-3p	PONS vs TCTX	
			hsa-miR-487a	hsa-miR-9-3p	hsa-miR-222-3p
			hsa-miR-202-3p	hsa-miR-302a-3p	hsa-miR-125b-5p
			hsa-miR-432-3p	hsa-miR-410	hsa-miR-328
			hsa-miR-495	hsa-miR-487b	hsa-miR-581
			hsa-miR-504	**hsa-miR-630**	hsa-miR-661
			hsa-miR-505-3p		
			hsa-miR-563		
			hsa-miR-578		
			hsa-miR-630		
			hsa-miR-668		

References

1. Cai, Y., Yu, X., Hu, S., et al.: A Brief Review on The Mechanisms of Mirna Regulation. Genomics, Proteomics & Bioinformatics 7, 147–154 (2009)
2. Wienholds, E., Kloosterman, W.P., Miska, E., et al.: MicroRNA Expression in Zebrafish Embryonic Development. Science 309, 310–311 (2005)
3. Wang, D., Qiu, C., Zhang, H., et al.: Human Microrna Oncogenes and Tumor Suppressors Show Significantly Different Biological Patterns: From Functions to Targets. PLoS ONE 5 e13067 (2010)
4. Wittmann, J., Jack, H.M.: Serum MicroRNAs as Powerful Cancer Biomarkers. Biochim. Biophys. Acta 1806, 200–207 (2010)
5. Anokye-Danso, F., Trivedi, C.M., Juhr, D., et al.: Highly Effcient miRNA-mediated Reprogramming of Mouse and Human Somatic Cells to Pluripotency. Cell Stem Cell 8, 376–388 (2011)
6. Krichevsky, A.M., King, K.S., Donahue, C.P., et al.: A MicroRNA Array Reveals Extensive Regulation of MicroRNAs During Brain Development. RNA 9, 1274–1281 (2003)
7. Schratt, G.M., Tuebing, F., Nigh, E.A., et al.: A Brain-specific MicroRNA Regulates Dendritic Spine Development. Nature 439, 283–289 (2006)
8. Miska, E., Alvarez-Saavedra, E., Townsend, M., et al.: Microarray Analysis of MicroRNA Expression in the Developing Mammalian Brain. Genome biology 5 R68 (2004)

9. Shao, N.Y., Hu, H.Y., Yan, Z., et al.: Comprehensive Survey of Human Brain MicroRNA by Deep Sequencing. BMC Genomics 11, 409 (2010)
10. Shahab, S., Matyunina, L., Hill, C., et al.: The Effects of MicroRNA Transfections on Global Patterns of Gene Expression in Ovarian Cancer Cells Are Functionally Coordinated. BMC Medical Genomics 5, 33 (2012)
11. Suzuki, M.M., Bird, A.: DNA Methylation Landscapes: Provocative Insights from Epigenomics. Nat. Rev. Genet. 9, 465–476 (2008)
12. Palmisano, W.A., Divine, K.K., Saccomanno, G., et al.: Predicting Lung Cancer by Detecting Aberrant Promoter Methylation in Sputum. Cancer Res. 60, 5954–5958 (2000)
13. Maruyama, R., Toyooka, S., Toyooka, K.O., et al.: Aberrant Promoter Methylation Profile of Bladder Cancer and Its Relationship to Clinicopathological Features. Cancer Res. 61, 8659–8663 (2001)
14. Farthing, C.R., Ficz, G., Ng, R.K., et al.: Global Mapping of Dna Methylation in Mouse Promoter Reveals Epigenetic Reprogramming of Pluripotency Genes. PLoS Genet 4 e1000116 (2008)
15. Su, Z., Xia, J., Zhao, Z.: Functional Complementation Between Transcriptional Methylation Regulation and Post-transcriptional microRNA Regulation in The Human Genome. BMC Genomics 12 S15 (2011)
16. Taguchi, Y-h.: Competitive Target Gene Regulation by Promoter Methylation and miRNA. IPSJ SIG Technical Reports 2012:1, 1–6 (2012)
17. Taguchi, Y.-H.: Inference of The Target Gene Regulation by miRNA via Mirage Server. In: Wan, J. (ed.) Introduction to Genetics – DNA Methylation and Gene Regulation, iConcept Press, Hong Kong (in press 2013)
18. Yao, M.J., Chen, G., Zhao, P.P., et al.: Transcriptome Analysis of MicroRNAs in Developing Cerebral Cortex of Rat. BMC Genomics 13, 232 (2012)
19. Babenko, O., Kovalchuk, I., Metz, G.A.: Epigenetic Programming of Neurodegenerative Diseases by An Adverse Environment. Brain Res 1444, 96–111 (2012)
20. Roshan, R., Ghosh, T., Scaria, V., et al.: MicroRNAs: Novel Therapeutic Targets in Neurodegenerative Diseases. Drug Discov. Today 14, 1123–1129 (2009)

Clustering and Assembling Large Transcriptome Datasets by EasyCluster2

Vitoantonio Bevilacqua[1,*], Nicola Pietroleonardo[1], Ely Ignazio Giannino[1], Fabio Stroppa[1], Graziano Pesole[2,3], and Ernesto Picardi[2,3]

[1] DEI - Politecnico di Bari - Via Orabona, 4, Bari, 70125, Italy
[2] DBBB - University of Bari, Bari
[3] Istituto di Biomembrane e Bioenergetica del Consiglio Nazionale delle Ricerche, Bari
bevilacqua@poliba.it

Abstract. EasyCluster is a well-established python software appropriately developed to produce reliable clusters by expressed sequence tags (EST) in order to infer and improve gene structures as well as discover potential alternative splicing events. In the present work we present EasyCluster2, a reimplementation of EasyCluster in Java programming language, able to manage genome scale transcriptome data produced by Roche 454 sequencers. EasyCluster2 has been developed to speed up the creation of gene-oriented clusters and facilitate downstream analyses as the assembly of full-length transcripts. In addition, EasyCluster2 can employ known annotations to refine the overall clustering procedure, embeds the AStalavista software to predict the impact of alternative splicing per cluster and provides output files in specific formats to be uploaded in the UCSC genome browser for an easy browsing of results. Thanks to the user-friendly interface, EasyCluster2 simplifies the interpretation of findings to researchers with no specific skills in bioinformatics. Easycluster2 executable is freely available at https://code.google.com/p/easycluster2/.

Keywords: EasyCluster2, expressed sequence tags, 454 reads, alternative splicing.

1 Introduction

Expressed sequence tags (ESTs) and full-length cDNAs (FL-cDNAs) are an invaluable source of evidence to infer reliable gene structures and discover potential alternative splicing events(1). Their biological potential can be fully exploited through EST clustering in which expressed sequences are linked to their specific gene loci of origin. To generate reliable gene-oriented clusters of ESTs, we developed the program EasyCluster that resulted the most accurate when compared to the state of the art software in this field (2-4). Nowadays, thanks to technological advances, EST-like sequences can be produced by pyrosequencing using the Roche 454 platform. Indeed, this is the only technology able to generate, through the GS FLX Titanium chemistry,

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2013, CCIS 375, pp. 231–236, 2013.

sequence reads up to 1 Kb long (http://www.454.com/) (5). Handling huge amount of EST-like data is extremely useful to detect alternative isoforms, improve gene annotations or simply create gene-oriented clusters for expression studies. Since EST-like data provide a fragmented overview of their genomic loci of origin, transcript assembly may be an optimal solution to annotate user-produced sequences. To benefit from long transcriptome reads, we developed EasyCluster2, a new version of EasyCluster, that can now manage genome scale transcriptome data and generate reliable gene-oriented clusters from 454 reads, facilitating downstream analyses and enabling the assembly of full-length transcripts. EasyCluster2 accepts as input read alignment files in GFF3 format (http://www.sequenceontology.org/gff3.shtml) generated by various alignment programs such as GMAP (6), refines the EST clustering using information of shared splice sites, and resolves potential mapping errors at exon-exon junctions using dynamic programming. In addition, EasyClusters2 can now handle unspliced ESTs (prominent in classical 454 data) and optimize the cluster definition with known gene annotations. A graph-based approach is used to assemble full-length transcripts belonging to a specific cluster, thus simplifying the investigation of post-transcriptional events as alternative splicing. Indeed, the AStalavista (7) program has been integrated in our tool allowing a quick way to explore alternative splicing without known reference transcripts. EasyCluster2 has been written in Java programming language with an easy graphical interface to simplify genome-level analyses to researchers not fully skilled in bioinformatics. The main EasyCluster executable is freely available at the following Google code page: https://code.google.com/p/easycluster2/

2 Results

Overview of EasyCluster2
The clustering procedure implemented in EasyCluster2 is summarized in the following nine steps:

1. An individual alignment file in GFF3 format is provided as input and parsed in memory exploiting JAVA classes of a custom library. Then reads are grouped according to their ‘exon’ features included in the GFF3 file;
2. Initial clusters are generated by overlapping genomic coordinates;
3. Refined Clusters are then produced using to the biological criterion of splice site sharing;
4. Potential mapping errors are corrected by an ad-hoc re-alignment strategy;
5. Unspliced (intronless) and Mixed (multi-mapping) reads are included in relevant clusters using proper criteria;
6. Clusters can be merged to take into account the fragmented locus sequencing;
7. Known annotations (if available) can be exploited to improve clusters correctness;
8. Full-length transcripts are assembled from generated clusters by a graph-based procedure;
9. Alternative splicing events can be predicted using the embedded AStalavista module.

Parsing of GFF3 Input Files
The clustering is performed chromosome by chromosome according to read alignments in the input GFF3 file. During this step, transcript orientation (read strand)

is also taken into account. Along the parsing, read alignments are classified in Unique (occurring in only one genome location) and Mixed (mapping on multiple genome locations). In addition, reads are further divided into Spliced (including at least 1 intron) and Unspliced (intronless). In the first clustering procedure only Unique and Spliced sequences are used. In the parsing step EasyCluster2 reads the input GFF3 file leading to the creation of appropriated and dedicated data structures. Reads alignments are finally coordinate sorted to speed up the creation of initial clusters.

First Clustering and Second Clustering and Refinements

The clustering procedure begins by instantiating an object representing the cluster. Such object is part of a custom library and its start and end coordinates are set-up to ones of the first read included into the cluster. For each read belonging to the read set to be clustered, EasyCluster2 verifies if its start coordinate is smaller than the cluster end coordinate. If this condition is satisfied the read is added to the corresponding cluster. This procedure is performed on the overall set of ESTs sorted by coordinates in ascending order. After the generation of initial clusters, called also pseudo-Clusters, EasyCluster2 refines each read group according to the biological criterion of splice site sharing. In this step, EasyCluster2 keeps track of all reads having common coordinates at exon level, excluding start or end coordinates of initial or terminal exons.

Since input reads are generally aligned individually, potential mapping errors may be present. For this reason we implemented a simple strategy to mitigate the effect of such problem following the idea of *Cluster Profiles*, consisting in an ordered list of read donors and acceptors for each cluster as well as their occurrences. In particular, the error correction at splice sites makes use of the Smith-Waterman algorithm (9) in the region surrounding each splice site (Fig. 1.a.).

Fig. 1.a. On first image, donor splicing site on the second exon is in line with a donor of the Cluster Profile belonging to the right neighborhood searched: part of the next exon is shifted to the first exon. On second image, donor splicing site on the first exon is in line with a donor of the Cluster Profile belonging to the left neighborhood searched: part of this exon is shifted to the next exon.

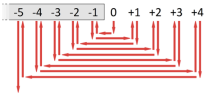


Fig. 1.b. Research in neighborhood

From a computational point of view the goal of this step of the algorithm is to check the shift of a read substring (or rather a portion of its exons) considering all Splicing Sites as annotated in the *Cluster Profile*. EasyCluster2 works as follow:

- For all couples of donor (*i-th* exon) and acceptor (*i+1-th* exon) sites of each EST (as thus, for each Exon) the algorithm verifies a potential correspondence; in other words, it checks if a specific coordinate of each EST Exon under analysis is present in the *Cluster Profile*.
- This check is done in a region surrounding the splicing site consisting in 10 nucleotides, 5 upstream and 5 downstream, respectively (Fig. 1.b.).

- Then the algorithm verifies if there is a correspondence of coordinates in the *Profile* for each coordinate of nucleotide in neighborhood, testing if coordinates "*splicing_site_+1", "splicing_site_+2", "splicing_site_+3"* (and so on) belong to the *Profile*.
- If a correspondence is found, the Smith-Waterman algorithm is used to verify the quality of the alignment onto the corresponding genomic region. The two exons under investigation are cut according to Smith-Waterman results and ready to be shifted to the first or second exon as shown in Fig. 1.a.

Inclusion of Unspliced and Mixed Reads

Unspliced reads, for which the orientation is indeterminate, are included in already generated clusters if completely comprised in exonic regions. Alternatively, Smith-Waterman is applied to facilitate the assessment of reads if any.

Unspliced reads that cannot be included in existent clusters, will create new independent clusters.

Mixed reads, instead, can be optionally inserted in clusters according to a *membership coefficient*, calculated by the following formula:

$$mc = \frac{nESTmapSS}{totSScluster}$$

Where ***totSScluster*** is the number of splice sites in the examined Cluster and ***nESTmapSS*** is the number of examined Cluster reads which have the same Mixed read splice sites.

Mixed reads are finally assigned to the cluster with the highest *membership coefficient.*

Transcript Assembly

A great novel of EasyCluster2 is the assembly of clustered reads in full-length transcripts by an algorithm based on the graph theory.

This process is accomplished by solving the inclusion and/or extension between pairwise reads, verifying the sharing of splice sites (Fig. 2.a/b/c).

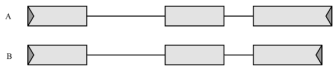


Fig. 2.a. Inclusion: every intron is shared and B is smaller than A

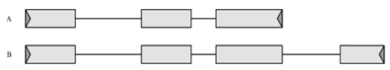


Fig. 2.b. Extension: every intron is shared and B is greater than A

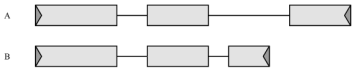


Fig. 2.c. No relationship: only the first intron is shared

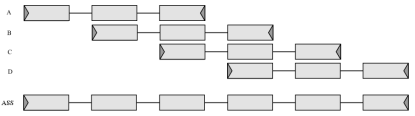


Fig. 2.d. Assembly result

Considering an ordered pair of reads we may have:

- Inclusion, if both reads share all splice sites and the size of the second read is smaller than the size of the first one;
- Extension, if both reads share splice sites and the size of the second reads is greater than the size of the first one;

- No relationship, if both reads have discordant splice sites.

According to above rules, EasyCluster2 performs transcript assembly. A graphical overview is shown in Fig. 2.d. below.

Alternative Splicing
The prediction of alternative splicing is a basic step once full-transcripts have been generated per each cluster. Although many programs tend to predict alternative splicing events defining a reference transcript and then valuating differences in the pattern of splice sites, EasyCluster2 takes a diverse approach. Indeed, it implements the AStalavista (5) program in which splice site inconsistencies and, thus, alternative splicing events are detected by looking at genomic coordinates without any reference transcript. This is an optimal solution in case in which gene annotations are not well known and no reference transcripts can be defined unambiguously.

Performance
EasyCluster2 accuracy was checked on our standard benchmark dataset comprising 111 human genes and 17,733 ESTs. In such benchmark we know exactly the relationship between each gene locus and corresponding ESTs of origin. Easycluster2 was able to correctly predict all 111 clusters outperforming the previous Easycluster implementation. In addition, we tested our software on a second human dataset including 21,599 transcripts from chromosome 21. This dataset was used to compare the performance of a recently released program named RCDA with the first version of EasyCluster. In particular, RCDA predicted 379 clusters while EasyCluster produced 389 in 90 min. In contrast, Easycluster2 generated 354 clusters compatibles with current annotations in UCSC in less than 10 min using default parameter or less than 40 min activating the exon refinement. Full-length transcripts were reconstructed in minutes and appeared consistent with RefSeq annotations included in the dataset. The consistency was estimated looking at shared introns since upstream and downstream transcript regions may differ.

3 Materials and Methods

EasyCluster2 has been developed in Java programming language and tested on unix based machine equipped with 2 quad core CPUs and 16GB of RAM. Datasets to evaluate each step of clustering procedure were simulated by 454sim (8) software. Our established benchmark dataset (2) was used to evaluate cluster accuracy.

4 Conclusions

EasyCluster2 is a reimplementation of EasyCluster software devoted to the generation of gene-oriented clusters by massive transcriptome reads. Our software is written in Java language and implements different novelties including a procedure to mitigate mapping errors at splice sites and an ad hoc solution to assemble full-length transcripts per cluster. In addition, EasyCluster2 can now predict alternative splicing events thanks to the embedded AStalavista module. Given the explosion of next generation sequencing and the concomitant increment of read lengths, we think that a tool as EasyCluster2 may be extremely useful for large-scale transcriptome experiments enabling complex genomic analyses to researchers not fully skilled in bioinformatics by employing an user-friendly interface.

5 Future Plans

Preliminary results on benchmark data sets (described above) suggest high accuracy of EasyCluster2 in producing effective clusters as well as reliable full-length transcripts. Given the importance of such kind of software in analysing huge amount of reads by the 454 platform, we will test extensively EasyCluster2 on both simulated and real datasets. In case of simulated data, we will employ known RefSeq annotations to produce EST-like reads under the 454 error model to evaluate the clustering procedure as well as the reconstruction of full isoforms. Such data will indispensable to expand the capabilities of EasyCluster2 in calculating gene expression differences among diverse experimental conditions. In addition, thank to technological improvements, we will try to apply EasyCluster2 on datasets from Illumina MiSeq machines able to generate millions of paired-end reads 250 nt long. In this way a method to include paired-end reads will also be implemented in EasyCluster2 as well as the possibility to use as input read alignments in the standard SAM/BAM format.

Acknowledgements. Authors thank Dr. Michael Sammeth and his lab for help in embedding AStalavista code in EasyCluster2. This work was supported by "Ministero dell'Istruzione, Università e Ricerca" (MIUR, Italy); Italian Ministry for Foreign Affairs (Italy-Israel Actions).

References

1. Nagaraj, S.H., Gasser, R.B., Ranganathan, S.: A hitchhiker's guide to expressed sequence tag (EST) analysis. Briefings in Bioinformatics 8, 6–21 (2007)
2. Picardi, E., Mignone, F., Pesole, G.: EasyCluster: a fast and efficient gene-oriented clustering tool for large-scale transcriptome data. BMC Bioinformatics 10, S10 (2009)
3. Picardi, E., Bevilacqua, V., Stroppa, F., Pesole, G.: An improved procedure for clustering and assembly of large transcriptome data. EMBnet. journal (2012)
4. Bevilacqua, V., Stroppa, F., Saladino, S., Picardi, E.: A novel approach to clustering and assembly of large-scale roche 454 transcriptome data for gene validation and alternative splicing analysis. In: Huang, D.-S., Gan, Y., Premaratne, P., Han, K. (eds.) ICIC 2011. LNCS, vol. 6840, pp. 641–648. Springer, Heidelberg (2012)
5. Droege, M., Hill, B.: The Genome Sequencer FLX System–longer reads, more applications, straightforward bioinformatics and more complete data sets. J. Biotechnol. 31, 136(1-2), 3–10 (2008)
6. Wu, T.D., Watanabe, C.K.: GMAP: a genomic mapping and alignment program for mRNA and EST sequences. Bioinformatics 21, 1859–1875 (2005)
7. Foissac, S., Sammeth, M.: ASTALAVISTA: dynamic and flexible analysis of alternative splicing events in custom gene datasets. Nucleic Acids Res 35, W297–W299 (2007)
8. Lysholm, F., Andersson, B., Persson, B.: An efficient simulator of 454 data using configurable statistical models. BMC Res Notes 4(1), 449 (2011)
9. Moustafa, A.: JAligner: Open source Java implementation of Smith-Waterman., `http://jaligner.sourceforge.net` (the date accessed)

Author Index