Tingwen Huang
Zhigang Zeng
Chuandong Li
Chi Sing Leung (Eds.)

LNCS 7665

Neural Information Processing

19th International Conference, ICONIP 2012
Doha, Qatar, November 2012
Proceedings, Part III

3 Part III

Springer

Lecture Notes in Computer Science 7665

Tingwen Huang Zhigang Zeng
Chuandong Li Chi Sing Leung (Eds.)

Neural Information Processing

19th International Conference, ICONIP 2012
Doha, Qatar, November 12-15, 2012
Proceedings, Part III

Springer

Volume Editors

Tingwen Huang
Texas A&M University at Qatar, Education City
P.O. Box 23874, Doha, Qatar
E-mail: tingwen.huang@qatar.tamu.edu

Zhigang Zeng
Huazhong University of Science and Technology
Department of Control Science and Engineering
1037 Luoyu Road, Wuhan, Hubei 430074, China
E-mail: zgzeng@gmail.com

Chuandong Li
Chongqing University, College of Computer Science
174 Shazhengjie Street, Chongqing 400044, China
E-mail: licd@cqu.edu.cn

Chi Sing Leung
City University of Hong Kong, Department of Electronic Engineering
83 Tat Chee Avenue, Kowloon, Hong Kong, China
E-mail: eeleungc@cityu.edu.hk

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-34486-2 e-ISBN 978-3-642-34487-9
DOI 10.1007/978-3-642-34487-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012949896

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

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

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

Printed on acid-free paper

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

Preface

This volume is part of the five-volume proceedings of the 19th International Conference on Neural Information Processing (ICONIP 2012), which was held in Doha, Qatar, during November 12–15, 2012. ICONIP is the annual conference of the Asia Pacific Neural Network Assembly (APNNA). This series of conferences has been held annually since 1994 and has become one of the premier international conferences in the areas of neural networks.

Over the past few decades, the neural information processing community has witnessed tremendous efforts and developments from all aspects of neural information processing research. These include theoretical foundations, architectures and network organizations, modeling and simulation, empirical study, as well as a wide range of applications across different domains. Recent developments in science and technology, including neuroscience, computer science, cognitive science, nano-technologies, and engineering design, among others, have provided significant new understandings and technological solutions to move neural information processing research toward the development of complex, large-scale, and networked brain-like intelligent systems. This long-term goal can only be achieved with continuous efforts from the community to seriously investigate different issues of the neural information processing and related fields. To this end, ICONIP 2012 provided a powerful platform for the community to share their latest research results, to discuss critical future research directions, to stimulate innovative research ideas, as well as to facilitate multidisciplinary collaborations worldwide.

ICONIP 2012 received tremendous submissions authored by scholars coming from 60 countries and regions across six continents. Based on a rigorous peer-review process, where each submission was evaluated by at least two reviewers, about 400 high-quality papers were selected for publication in the prestigious series of *Lecture Notes in Computer Science*. These papers cover all major topics of theoretical research, empirical study, and applications of neural information processing research. In addition to the contributed papers, the ICONIP 2012 technical program included 14 keynote and plenary speeches by Majid Ahmadi (University of Windsor, Canada), Shun-ichi Amari (RIKEN Brain Science Institute, Japan), Guanrong Chen (City University of Hong Kong, Hong Kong), Leon Chua (University of California at Berkeley, USA), Robert Desimone (Massachusetts Institute of Technology, USA), Stephen Grossberg (Boston University, USA), Michael I. Jordan (University of California at Berkeley, USA), Nikola Kasabov (Auckland University of Technology, New Zealand), Juergen Kurths (University of Potsdam, Germany), Erkki Oja (Aalto University, Finland), Marios M. Polycarpou (University of Cyprus, Cyprus), Leszek Rutkowski (Technical University of Czestochowa, Poland), Ron Sun (Rensselaer Polytechnic Institute, USA), and Jun Wang (Chinese University of Hong Kong, Hong Kong). The

ICONIP technical program included two panels. One was on “Challenges and Promises in Computational Intelligence” with panelists: Shun-ichi Amari, Leon Chua, Robert Desimone, Stephen Grossberg and Michael I. Jordan; the other one was on “How to Write Better Technical Papers for International Journals in Computational Intelligence” with panelists: Derong Liu (University of Illinois of Chicago, USA), Michel Verleysen (Université catholique de Louvain, Belgium), Deliang Wang (Ohio State University, USA), and Xin Yao (University of Birmingham, UK). The ICONIP 2012 technical program was enriched by 16 special sessions and “The 5$^{\text{th}}$ International Workshop on Data Mining and Cybersecurity.” We highly appreciate all the organizers of special sessions and workshop for their tremendous efforts and strong support.

Our conference would not have been successful without the generous patronage of our sponsors. We are most grateful to our platinum sponsor: *United Development Company PSC (UDC)*; gold sponsors: *Qatar Petrochemical Company, ExxonMobil* and *Qatar Petroleum*; organizers/sponsors: *Texas A&M University at Qatar* and *Asia Pacific Neural Network Assembly.* We would also like to express our sincere thanks to the IEEE Computational Intelligence Society, International Neural Network Society, European Neural Network Society, and Japanese Neural Network Society for technical sponsorship.

We would also like to sincerely thank Honorary Conference Chair Mark Weichold, Honorary Chair of the Advisory Committee Shun-ichi Amari, the members of the Advisory Committee, the APNNA Governing Board and past presidents for their guidance, the Organizing Chairs Rudolph Lorentz and Khalid Qaraqe, the members of the Organizing Committee, Special Sessions Chairs, Publication Committee and Publicity Chairs, for all their great efforts and time in organizing such an event. We would also like to take this opportunity to express our deepest gratitude to the members of the Program Committee and all reviewers for their professional review of the papers. Their expertise guaranteed the high quality of the technical program of the ICONIP 2012!

We would like to express our special thanks to Web manager Wenwen Shen for her tremendous efforts in maintaining the conference website, the publication team including Gang Bao, Huanqiong Chen, Ling Chen, Dai Yu, Xing He, Junjian Huang, Chaobei Li, Cheng Lian, Jiangtao Qi, Wenwen Shen, Shiping Wen, Ailong Wu, Jian Xiao, Wei Yao, and Wei Zhang for spending much time to check the accepted papers, and the logistics team including Hala El-Dakak, Rob Hinton, Geeta Megchiani, Carol Nader, and Susan Rozario for their strong support in many aspects of the local logistics.

Furthermore, we would also like to thank Springer for publishing the proceedings in the prestigious series of *Lecture Notes in Computer Science.* We would, moreover, like to express our heartfelt appreciation to the keynote, plenary, panel, and invited speakers for their vision and discussions on the latest research developments in the field as well as critical future research directions, opportunities, and challenges. Finally, we would like to thank all the speakers, authors, and participants for their great contribution and support that made ICONIP 2012 a huge success.

November 2012

Tingwen Huang
Zhigang Zeng
Chuandong Li
Chi Sing Leung

Organization

Honorary Conference Chair

Mark Weichold	Texas A&M University at Qatar, Qatar

General Chair

Tingwen Huang	Texas A&M University at Qatar, Qatar

Program Chairs

Andrew Leung	City University of Hong Kong, Hong Kong
Chuandong Li	Chongqing University, China
Zhigang Zeng	Huazhong University of Science and Technology, China

Advisory Committee

Honorary Chair

Shun-ichi Amari	RIKEN Brain Science Institute, Japan

Members

Majid Ahmadi	University of Windsor, Canada
Sabri Arik	Istanbul University, Turkey
Salim Bouzerdoum	University of Wollongong, Australia
Jinde Cao	Southeast University, China
Jonathan H. Chan	King Mongkut's University of Technology, Thailand
Guanrong Chen	City University of Hong Kong, Hong Kong
Tianping Chen	Fudan University, China
Kenji Doya	Okinawa Institute of Science and Technology, Japan
Wlodzislaw Duch	Nicolaus Copernicus University, Poland
Ford Lumban Gaol	Bina Nusantara University, Indonesia
Tom Gedeon	Australian National University, Australia
Stephen Grossberg	Boston University, USA
Haibo He	University of Rhode Island, USA
Akira Hirose	University of Tokyo, Japan
Nikola Kasabov	Auckland University of Technology, New Zealand

Irwin King	The Chinese University of Hong Kong, Hong Kong
James Kwow	Hong Kong University of Science and Technology, Hong Kong
Soo-Young Lee	Advanced Institute of Science and Technology, Korea
Xiaofeng Liao	Chongqing University, China
Chee Peng Lim	Universiti Sains Malaysia, Malaysia
Derong Liu	University of Illinois at Chicago, USA
Bao-Liang Lu	Shanghai Jiao Tong University, China
John MacIntyre	University of Sunderland, UK
Erkki Oja	Helsinki University of Technology, Finland
Nikhil R. Pal	Indian Statistical Institute, India
Marios M. Polycarpou	University of Cyprus, Cyprus
Leszek Rutkowski	Czestochowa University of Technology, Poland
Noboru Ohnishi	Nagoya University, Japan
Ron Sun	Rensselaer Polytechnic Institute, USA
Ko Sakai	University of Tsukuba, Japan
Shiro Usui	RIKEN, Japan
Xin Yao	University of Birmingham, UK
DeLiang Wang	Ohio State University, USA
Jun Wang	Chinese University of Hong Kong, Hong Kong
Li-Po Wang	Nanyang Technological University, Singapore
Rubin Wang	East China University of Science and Technology, China
Zidong Wang	Brunel University, UK
Huaguang Zhang	Northeastern University, China

Organizing Committee

Chairs

Rudolph Lorentz	Texas A&M University at Qatar, Qatar
Khalid Qaraqe	Texas A&M University at Qatar,Qatar

Members

Hassan Bazzi	Texas A&M University at Qatar, Qatar
Hala El-Dakak	Texas A&M University at Qatar, Qatar
Mohamed Elgindi	Texas A&M University at Qatar, Qatar
Jihad Mohamad Jaam	Qatar University, Qatar
Samia Jones	Texas A&M University at Qatar, Qatar
Uvais Ahmed Qidwai	Qatar University, Qatar
Paul Schumacher	Texas A&M University at Qatar, Qatar

Special Sessions Chairs

Zijian Diao	Ohio University, USA
Hassab Elgawi Osman	The University of Tokyo, Japan
Paul Pang	Unitec Institute of Technology, New Zealand

Publicity Chairs

Mehdi Roopaei	Shiraz University, Iran
Enchin Serpedin	Texas A&M University,USA
Maolin Tang	Queensland University of Technology, Australia

Program Committee Members

Sabri Arik
Emili Balaguer Ballester
Gang Bao
Matthew Casey
Li Chai
Jonathan Chan
Mou Chen
Yangquan Chen
Mingcong Deng
Ji-Xiang Du
El-Sayed El-Alfy
Osman Elgawi
Peter Erdi
Wai-Keung Fung
Yang Gao
Erol Gelenbe
Nistor Grozavu
Ping Guo
Fei Han
Hanlin He
Shan He
Bin He
Jinglu Hu
He Huang
Kaizhu Hunag
Jihad Mohamad Jaam
Minghui Jiang
Hu Junhao
John Keane
Sungshin Kim
Irwin King
Sid Kulkarni
H.K. Kwan
James Kwok
Wk Lai
James Lam
Soo-Young Lee
Chi Sing Leung
Tieshan Li
Bin Li
Yangmin Li
Bo Li
Ruihai Li
Hai Li
Xiaodi Li
Lizhi Liao
Chee-Peng Lim
Ju Liu
Honghai Liu
Jing Liu
C.K. Loo
Luis Martínez López
Wenlian Lu
Yanhong Luo
Jinwen Ma
Mufti Mahmud
Jacek Mańdziuk
Muhammad Naufal Bin Mansor
Yan Meng
Xiaobing Nie
Sid-Ali Ouadfeul
Seiichi Ozawa
Shaoning Paul Pang
Anhhuy Phan
Uvais Qidwai
Ruiyang Qiu
Hendrik Richter
Mehdi Roopaei
Thomas A. Runkler
Miguel Angel Fernández Sanjuán
Ruhul Sarker
Naoyuki Sato
Qiankun Song
Jochen Steil

John Sum
Bing-Yu Sun
Norikazu Takahashi
Kay Chen Tan
Ying Tan
Maolin Tang
Jinshan Tang
Huajin Tang
H. Tang
Ke Tang
Peter Tino
Haifeng Tou
Dat Tran
Michel Verleysen
Dan Wang
Yong Wang
Ning Wang
Zhanshan Wang
Xin Wang
Dianhui Wang
Ailong Wu
Bryant Wysocki
Bjingji Xu
Yingjie Yang
Shengxiang Yang
Wenwu Yu
Wen Yu
Xiao-Jun Zeng
Xiaoqin Zeng
Junping Zhang
Zhong Zhang
Wei Zhang
Jie Zhang
Dongbin Zhao
Hongyong Zhao
Huaqing Zhen

Publications Committee Members

Gang Bao
Guici Chen
Huangqiong Chen
Ling Chen
Shengle Fang
Lizhu Feng
Xing He
Junhao Hu
Junjian Huang
Feng Jiang
Bin Li
Chaobei Li
Yanling Li
Mingzhao Li
Lei Liu
Xiaoyang Liu
Jiangtao Qi
Wenwen Shen
Cheng Wang
Xiaohong Wang
Zhikun Wang
Shiping Wen
Ailong Wu
Yongbo Xia
Jian Xiao
Li Xiao
Weina Yang
Zhanying Yang
Wei Yao
Tianfeng Ye
Hongyan Yin
Dai Yu
Lingfa Zeng
Wei Zhang
Yongchang Zhang
Yongqing Zhao
Song Zhu

Platinum Sponsor

UDC

المتحدة للتنمية

UNITED DEVELOPMENT CO.

Gold Sponsors

QAPCO

ExxonMobil

Table of Contents – Part III

Session 3: Algorithms

Centroid Neural Network with Simulated Annealing and Its Application to Color Image Segmentation

Do-Thanh Sang, Dong-Min Woo*, and Dong-Chul Park

Dept. of Electronics Engineering, Myongji University, Korea 449-728
sang.dothanh@gmail.com, {dmwoo,parkd}@mju.ac.kr

Abstract. Centroid Neural Network (CNN) with simulated annealing is proposed and applied to a color image segmentation problem in this paper. CNN is essentially an unsupervised competitive neural network scheme and is a crucial algorithm to diminish the empirical process of parameter adjustment required in many unsupervised competitive learning algorithms including Self-Organizing Map. In order to achieve lower energy level during its training stage further, a supervised learning concept, called simulated annealing, is adopted. As a result, the final energy level of CNN with simulated annealing (CNN-SA) can be much lower than that of the original Centroid Neural Network. The proposed CNN-SA algorithm is applied to a color image segmentation problem. The experimental results show that the proposed CNN-SA can yield favorable segmentation results when compared with other conventional algorithms.

Keywords: Color image, Gray level, Segmentation, Centroid Neural Network.

1 Introduction

Image segmentation is the process of partitioning an image into a set of disjointed areas with uniform and homogeneous attributes such as intensity, color, tone or texture. A large number of different segmentation techniques have been developed to enable the object recognition and localization system to execute more accurately.

Thresholding is a widely used method in segmenting monochrome images. Bilevel thresholding method assigns a pixel to one class if its gray level is less than a specified threshold, and otherwise assigns it to the other classes [1]. Generally, one can select more than one threshold, and use these thresholds to separate the whole range of gray values into several sub ranges. This process is called multilevel thresholding. This method does not require prior information of the image and also has a low computation complexity. Nevertheless, it does not work well for an image without any obvious peaks or with broad and flat valleys. Moreover, it does not use any spatial information at all. Compared to monochrome images,

[*] Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 1–8, 2012.

color images are useful or even necessary in computer vision, because they provide additional information such as intensity. Color image processing thus is becoming more practical nowadays. Among many existing methods of color image segmentation, four main categories can be distinguished: pixel-based techniques, region-based techniques, contour-based techniques, and hybrid techniques. Unsupervised learning is widely applied in some applications, where the image features are unknown, such as nature scene understanding, satellite image analysis, etc. Many algorithms impose spatial constraints on clustering algorithms for segmenting image data. One of the most widely used algorithms employing fuzzy clustering techniques is the Fuzzy c-Means (FCM) [2], which has been proposed as an improvement on earlier clustering algorithms such as the Self-Organizing Map (SOM) [3] and the k-Means. Nevertheless, the FCM has the problem of exhaustive computational burden in classifying each pixel based on color feature space, especially for large images. Recently, Guo and Ming [4] have developed a hybrid technique that incorporates SOM and Simulated Annealing (SA) into color image segmentation. The SA is used to find optimal clusters from SOM prototypes; however, its drawbacks include the need for a lot of trial and error to obtain the optimal parameters, and it is very hard to implement such extensive trial and error.

In this paper, we adopt Centroid Neural Network (CNN) [5] to construct the ”natural grouping” of the image without using any prior knowledge. The CNN algorithm does not require a predetermined schedule for learning coefficient and a total number of iterations for clustering. We integrate a supervised learning concept, called simulated annealing into CNN to intensify the performance in order to achieve lower energy level during its training stage further. As a result, the final energy level of CNN with simulated annealing (CNN-SA) can be much lower than that of the original Centroid Neural Network. The effect of color image segmentation is dependent not only the algorithm but also on the color coordinate, hence the paper surveys RGB and L*u*v* coordination to obtain the best in color segmentation. We compare the segmentation results of natural scene images extracted from *Berkeley database* to show the effectiveness of the proposed method.

2 Choosing Color Information

Color is discerned by humans as a combination of tristimuli R (red), G (green), and B (blue), which are usually called the three primary colors. We can derive other kinds of color representations (spaces) by using either linear or nonlinear transformations from RGB representation. Several color spaces, such as RGB, CIE XYZ, HSI, L*u*v* are utilized in color image segmentation, but none of these can transcend the others for all kinds of color images. Selecting the best color space for all cases is still one of the difficulties in color image segmentation [6].

The RGB color space can be geometrically illustrated in a 3-dimensional cube. The coordinates of each point inside the cube represent the values of red, green and blue constituents, respectively. The RGB space is suitable for color display, but not good for color segmentation and analysis because of the high correlation among the R, G, and B components. High correlation means that if the intensity changes, these three components will change accordingly. Moreover, the measurement of a color in RGB space does not represent color differences in a uniform scale. Hence, it is impossible to evaluate the similarity of two colors from their distance in RGB space.

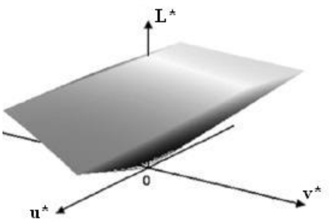


Fig. 1. L*u*v* color space represented in a 3-dimensional cube

Fig. 1 depicts color distributions in L*u*v*. It is concluded that the Modified CIE L*u*v* performs better than other color spaces. The CIE L*u*v* color space is defined from the CIE standard color model XYZ. L* is the luminance component, u* and v* are color components, the u* axis varies from green to red, and the v* axis changes from blue to yellow. The conversion from RGB to Modified CIE L*u*v* is a nonlinear transformation that includes the following two steps.

2.1 RGB to CIE XYZ

CIE XYZ can be transmuted from RGB by a 3x3 matrix transformation using Equation (1).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

2.2 CIE XYZ to Modified CIE L*u*v*

Intermediate quantities u' and v' can be computed using Equation (2).

$$u' = \frac{4X}{X + 15Y + 3Z} \quad v' = \frac{9Y}{X + 15Y + 3Z} \tag{2}$$

In [4], researchers used Modified CIE L*u*v* in lieu of standard CIE L*u*v* because the brightness L is proportional to $\sqrt{Y}$ rather than $\sqrt[3]{Y}$ for a complex viewing environment. Then, we have the relations, shown in Equation (3).

$$L^* = 10\sqrt{Y} \;\; u^* = 13L^*(u' - u'_n) \;\; v^* = 13L^*(v' - v'_n) \tag{3}$$

where$(u'_n, v'_n) = (0.1978,\, 0.4683)$ is used by default.

3 Application of Centroid Neural Network

3.1 Centroid Neural Network

The CNN algorithm originated from the conventional k-means algorithm finds the centroid of data in corresponding clusters at each presentation of the data vector. In lieu of calculating the centroids of the clusters while every piece of data is being presented, the CNN algorithm updates data weights only when the status of the output neuron for the presenting data has changed: that is, the weights of the winner neuron in the current epoch for the data change only when the winner neuron did not win the data in the previous presentation and the weights of the winner neuron in the previous epoch for the data change only when the neuron does not win the data in the current epoch. We call the former one a "winner neuron" and the latter one a "loser neuron". When an input vector x is applied to the network at time n, the weight update equations for winner neuron j and loser neuron i in CNN can be written as in Equations (4) and (5).

$$w_j(n+1) = \frac{1}{N_j+1}\left[N_j w_j(n) + x(n)\right] = w_j(n) + \frac{1}{N_j+1}\left[x(n) - w_j(n)\right] \quad (4)$$

$$w_i(n+1) = \frac{1}{N_i-1}\left[N_i w_i(n) - x(n)\right] = w_i(n) - \frac{1}{N_i-1}\left[x(n) - w_i(n)\right] \quad (5)$$

In Equations (4) and (5), $w_j(n)$ and $w_i(n)$ represent the weight vectors of the winner neuron and the loser neuron, respectively, while N_i and N_j denote the number of data vectors in cluster i^{th} and j^{th} at the time of iteration, respectively.

The learning rule for CNN is based on the following theorem and on the condition for minimum energy clustering:

- **Theorem 1:** The centroid of data in a cluster is the solution that gives minimum energy in L_2 norm.
- **Minimum energy condition:** The weights for a given output neuron should be chosen in such a way as to minimize the total distance in L_2 norm from the vectors in its class, such as

$$w_j = \min_w \sum_{i=1}^{N_j} \|x_j(i) - w\|^2 \quad (6)$$

Using Theorem 1, Equation (6) can be expressed as

$$w_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_j(i) \quad (7)$$

where N_j is the number of members in cluster j.

When CNN is compared with other conventional competitive learning algorithms, the CNN produces very comparable results with less computational effort. That is, the CNN requires neither a predetermined schedule for learning gain nor a total number of iterations for clustering; it converges stably to suboptimal solutions, while the conventional algorithms, including the Self Organizing Map (SOM), may give unstable results depending on the initial learning gain and the total number of iterations.

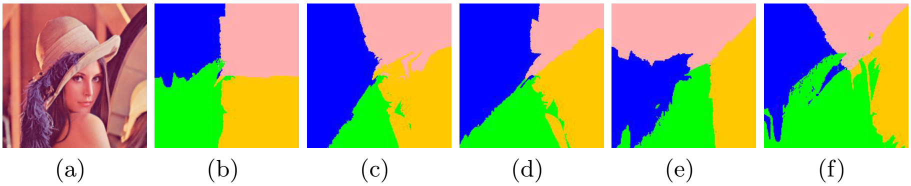


Fig. 2. Advantages of the CNN algorithm over other conventional algorithms in image segmentation example (a) LENA image; Segmented image by using: (b) CNN (c) FCM after first execution (d) FCM after second execution (e) SOM with after execution (f) SOM after second execution

Other modifications of this clustering technique are realized by increasing the dimensions of the feature space by introducing additional features, such as the geometrical coordinates of a pixel in the image. Fig. 2 is the result of segmenting the LENA color image by using L*u*v* and horizontal and vertical coordination. By visually comparing the segmented images, the image segmented by using CNN is more stable than that done by FCM and SOM, because the two other algorithms yield different results depending on the initial conditions.

3.2 CNN-SA Algorithm

The aim of Simulated Annealing (SA) is to seek the optimal clusters in terms of energy like most of unsupervised algorithms. The energy function (or cost function), E, in L_2 norm is defined as follows:

Find $\{w_i, 1 \leq i \leq M\}$ such that minimize

$$E = \sum_{i=1}^{M} E_i = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \|x_i(j) - w_i\|^2 \tag{8}$$

with $N = \sum_{i=1}^{M} N_i$

where M is the number of clusters.

The process of SA clustering is summarized as follows:

(1) $T = T_{max}$, distribute M data to K clusters, generate the initial energy J;
(2) While $(T > T_{min})$

```
For (i=1 to N) (N: the number of point redistribution)
    Transfer one point randomly from cluster $c_i$ to $c_j$;
    Compute the energy J';
    If ($J' < J$) Accept the new state;
    Else Reject the new state;
End for
Decrement T;
End while
```

We apply SA algorithm in the middle of CNN right before generating new cluster. Detail of CNN is stated in [5]. Applying SA after completed CNN is inefficient in some cases. Experiments present this circumstance.

4 Experiments

In the experiments, the configuration of SOM-SA is: initial learning rate: 0.7, total number of iterations: 500. SOM-SA are executed several times to get reasonable above parameters as well as results when compare with CNN-SA. In order to get fair results between algorithms, the parameters of SA are identical to all methods and set as follows. Maximum temperature (T_{max}): 100, minimum temperature (T_{min}): 90, annealing factor: 0.98, number of point redistribution: 200.

Table 1. Energy of algorithms

Image	cluster no.	Mixed CNN-SA	CNN-SA	SOM-SA

House	4	k=2: 27.83352E5
		k=3: 15.00047E5
		k=4: 11.15889E5
			28.83325E5	41.08434E5

Flower	3	k=2: 53.20481E5
		k=3: 47.10243E5
			49.13851E5	80.86287E5

Valley	3	k=2: 37.37489E5
		k=3: 15.00047E5
			39.48386E5	37.97711E5

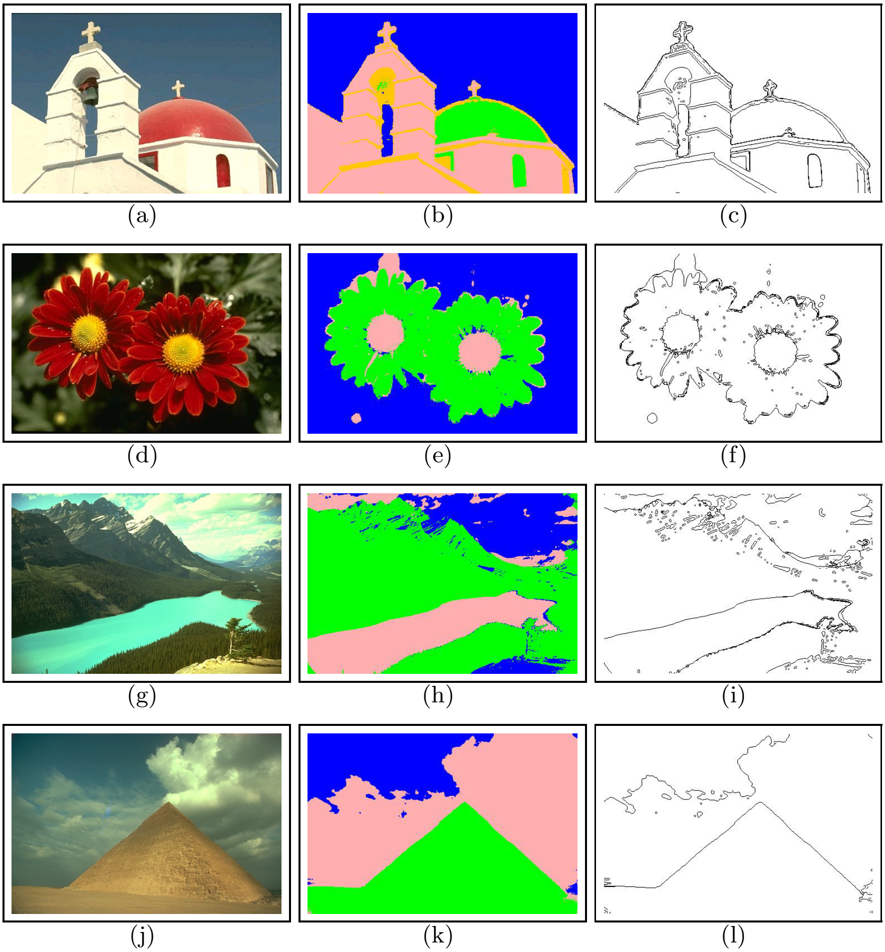
Pyramid	3	k=2: 28.97152E5
		k=3: 19.68023E5
			28.75196E5	28.11466E5

In Table 1, the energy in mixed CNN-SA algorithm always decreases along with going up of number of clusters (k).

We use the evaluation function in [7] to measure the segmentation results quantitatively as follows:

$$Q(I) = \frac{\sqrt{R}}{10000(W \times H)} \times \sum_{i=1}^{R} \left[\frac{e_i^2}{1 + \log A_i} \right] \tag{9}$$

where W x H is the image size, R is the number of regions of the segmented image, A_i is the area of the ith region, e_i is the sum of Euclidean distance



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 3. Color image segmentation of ”House”, ”Flower”, ”Valley” and ”Pyramid” images (a) Original ”House” image sized 481×321; (b) Segmented ”House” image by mixed CNN-SA; (c) Edge of segmented regions of ”House” image by mixed CNN-SA; (d) Original ”Flower” image sized 481×321; (e) Segmented ”Flower” image by mixed CNN-SA; (f) Edge of segmented regions of ”Flower” image by mixed CNN-SA; (g) Original ”Valley” image sized 481×321; (h) Segmented ”Valley” image by mixed CNN-SA; (i) Edge of segmented regions of ”Valley” image by mixed CNN-SA; (j) Original ”Pyramid” image sized 481×321; (k) Segmented ”Pyramid” image by mixed CNN-SA; (l) Edge of segmented regions of ”Pyramid” image by mixed CNN-SA

between the L*u*v* color vectors of the pixels of region i in original image and the color vector attributed to region i in the segmented image.

In Table 2, the quantitative errors of mixed CNN-SA algorithm are smaller than other algorithms. Fig. 3 depicts final segmented images, which show the proposed CNN-SA algorithm is a very efficient segmentation method.

Table 2. Quantitative error of algorithms

Image	SOM-SA	CNN-SA	Mixed CNN-SA
House	13.6063E-3	9.6539E-3	**2.0519E-3**
Flower	50.3777E-3	20.3523E-3	**16.1658E-3**
Valley	11.4188E-3	12.0573E-3	**7.427E-3**
Pyramid	5.8216E-3	6.9215E-3	**3.3516E-3**

5 Conclusions

This paper proposes an unsupervised learning algorithm for color image segmentation by using CNN-SA. The selection of proper parameters in SOM has been left in state of the art while CNN can significantly diminish the influences of the related parameters for SOM including the location of initial cluster centers and the number of iterations. In the color image segmentation, we employ L*u*v* space, which gives relatively better results. Since SOM-SA requires an intensive parameter tuning process to get the reasonable parameters, it can be concluded that the proposed CNN-SA algorithm is a very efficient segmentation method. SA's drawbacks include the need for a great deal of computing load for many runs and carefully chosen tunable parameters.

Acknowledgements. This work was supported by National Research Foundation of Korea Grant funded by the Korean Government (2012-004950).

References

1. Huang, L.K., Wang, M.J.: Image Thresholding by Minimizing the Measures of Fuzziness. Pattern Recogn. 28, 41–51 (1995)
2. Yang, J.F., Hao, S.S., Chung, P.C.: Color Image Segmentation Using Fuzzy C-Means and Eigenspace Projections. Signal Process. 82, 461–472 (2002)
3. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (1995)
4. Dong, G., Xie, M.: Color Clustering and Learning for Image Segmentation Based on Neural Networks. IEEE Trans. Neural Networks 16, 925–936 (2005)
5. Park, D.C.: Centroid Neural Network for Unsupervised Competitive Learning. IEEE Trans. Neural Networks 11, 520–528 (2000)
6. Cheng, H.D., Jiang, X.H., Sun, Y., Wang, J.: Color Image Segmentation: Advances and Prospects. Pattern Recogn. 34, 2259–2281 (2001)
7. Borsotti, M., Campadelli, P., Schettini, R.: Quantitiative Evaluation of Color Image Segmentation Results. Pattern Recogn. Lett. 19, 741–747 (1998)

Efficient Non-linear Filter for Impulse Noise Removal in Document Images

Ali Awad

Faculty of Engineering and Information Technology AL-Azhar university-Gaza, Palestine
aawad@alumni.stevens.edu

Abstract. A novel method is proposed in this paper to restore document images. The proposed method is based on finding the degree of similarity between the tested pixel and its neighbors in different door's sizes. If the tested pixel in every door size has enough similarity with at least two pixels, then the tested pixel is deemed original pixel. The number of two pixels is chosen, to make sure that the tested pixel is a part of an original text or a part in a series of similar pixels. Simulation results indicate that the new method delivers superior performance rapidly and efficiently either in terms of the noise removal or details preservation.

Keywords: Image denoising, Impulse noise, Non-linear filter.

1 Introduction

Data of documents that are taken from optical scanning or digital camera represents the image pixels. Many pre-processing tasks are performed on the attained pixels for further image analysis such as noise reduction to reduce extraneous data. Impulse noise besides the Gaussian noise is the common noises that may affect the document images. Thus, there is a demand to restore different corrupted documents images that may contain handwritten signatures, vehicle license numbers, handwritten-texts, and machine printed texts. To this end, many papers are proposed to tackle this problem. The average mean square error method is proposed by Meloche and Zamar [1] to remove Gaussian noise based on the content of the neighboring pixels. In [2] Hitchcock and Glasbey attempt to recover binary images of blob-like and filamentous objects from multilevel pixel values. Heanue, Gürkan and Hesselink [3] propose a non-linear recursive method, based on Veterbi algorithm with Decision Feedback to detect the binary data in the presence of inter-symbol interference. This technique sometimes suffers from error propagation.

Total variation minimizing models[4-6] are used by many authors to restore the document images. In [7] the authors proposed a convergent method to restore the corrupted binary images by finding global minimizer of the total-variation energy functional.

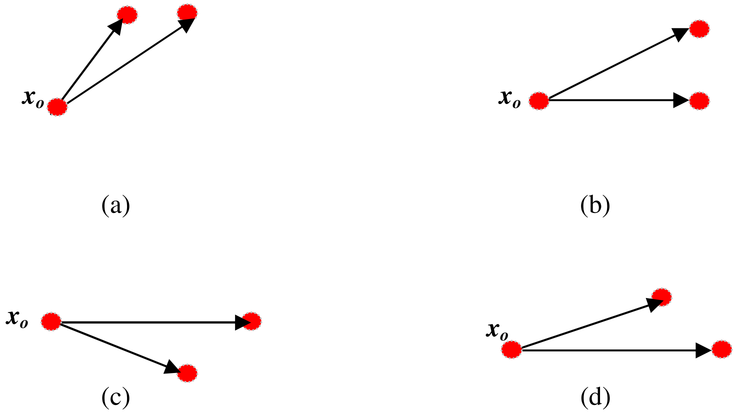
Besides the above techniques is thresholding technique. It is simple but it is effective method used to separate the objects from the image background. Thresholding techniques are implemented to extract printed characters, logo, and graphical content from the document images [8-11]. In [12] Mitra proposes an efficient method based on maximizing

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 9–16, 2012.

the ratio of the standard deviations of two different overlapping pixel clusters to restore document images corrupted by impulse and Gaussian noises. Among all the much known approaches is the mathematical morphological technique, which is used in the processing of geometrical structures. It is most commonly implemented on digital images in different research areas as image restoration, edge detection, and object recognition. The noise reduction methods mentioned in [13-16] to reduce the noise in document images are based on the mathematical morphological theory and they have achieved great success. However, Most of the previous methods are used for the removal of salt and pepper noise, Gaussian noise, or both. In the current paper, a different noise named as random valued impulse noise is used. Random-valued impulse noise is more difficult to detect, since it may take any values in the dynamic range of [0,255]. Images degraded by salt and pepper noise are restored as well. Note that, salt and pepper noise unlike the random noise in the sense that salt and pepper noise has two levels 0 and 255 but random valued impulse noise has 256 levels. During the simulation experiments we assumed that the random noise is distributed uniformly over the image and the values of the document images may take any value in the dynamic range. The restoration results show that the proposed method removes the noise efficiently and at the same time preserve the image details.

2 Algorithm Description

The goal of the proposed algorithm is to detect the noisy pixels and to estimate their original values. The proposed algorithm is based on: (1) The fact that the noisy pixels are random, and therefore they have random values. (2) The original pixels of the original text are similar in values which have a specific range of differences. (3) The original pixels are connected together in one or more directions. (4) Any tested pixel to be considered original pixel it should have similarity with at least two other consecutive pixels allied in one direction. The following shapes show the expected connections between the tested pixel and its surrounding.



(a) (b)

(c) (d)

Fig. 1. The expected connections between the tested pixel x_o, and two other surrounding pixels. Connections in a, b, c, and d are most probably occur between original pixels. Connections in e,f, and g are most probably happen between noisy pixels.

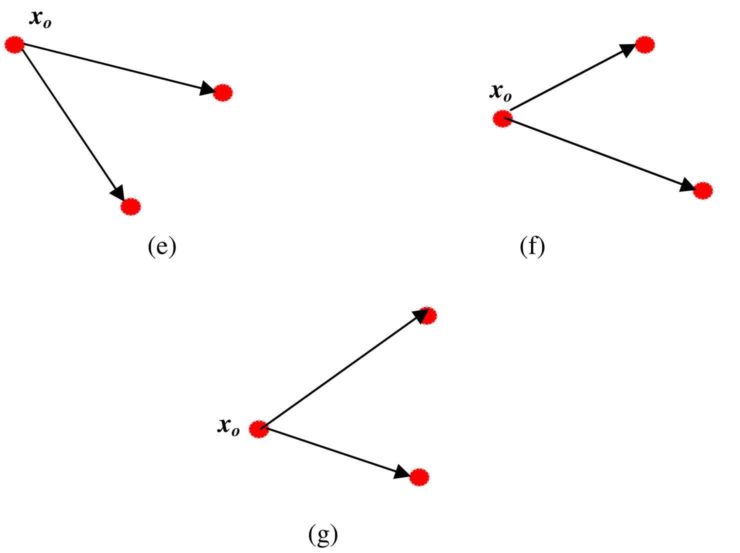


Fig. 1. (*continued*)

In shapes a, b, c and d, the tested pixel $\boldsymbol{x_o}$ is connected with two consecutive pixels. Shape (a) shows two pixels horizontally connected in the same row. Shape (b) shows two pixels vertically connected in the same column. Shapes (c) and (d) show two pixels each in different column are connected obliquely. Shapes (e) and (f) show two disconnected pixels in two non-consecutive columns. Shape (g) shows two disconnected pixels in the same column but in different rows.

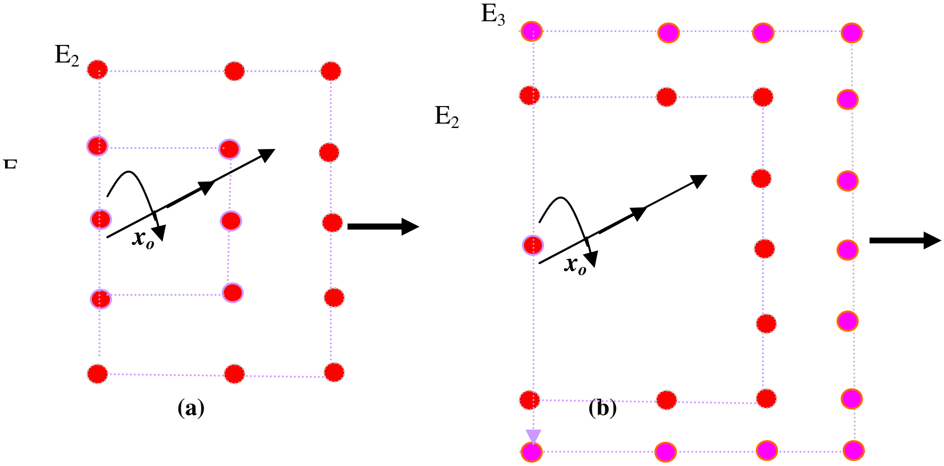


Fig. 2. Two doors each of two edges that are used in detecting the originality of the tested pixel $\boldsymbol{x_0}$. Fig. (a) has edge E$_1$ with 5 pixels and edge E$_2$ with 9 pixels. Fig. (b) has edge E$_2$ with 9 pixels and edge E$_3$ with 13 pixels.

The expected connections can be found in Fig.2 which looks like a door. From Fig.2, one can conclude that the tested pixel $\boldsymbol{x_0}$ that has connections with two disconnected (inconsequence) pixels is most probably noisy pixels. Also, there is a likelihood that the tested pixel in shapes a, b, c, and d in Fig.1 be connect with other two consecutive pixels but these pixels are noisy pixels, which leads to a false detection

For resolving these ambiguities, we pass the tested pixel into cascade of tests each includes a door of two edges and of different size. The pixels that satisfy the condition of the first door are tested again by using another door of higher size, as shown in fig.2. Pixels that satisfy the entire conditions of the different door sizes are considered original pixels. For describing this method, define the edges E_1 and E_2 of the door number ***i*** as:

$$E_1^i = \left\{ x_{11}^{di}, x_{21}^{di}, x_{31}^{di}, \ldots\ldots, x_{(5+4(i-1))1}^{di} \right\} \tag{1}$$

$$E_2^i = \left\{ x_{12}^{di}, x_{22}^{di}, x_{32}^{di}, \ldots\ldots, x_{(5+4i)2}^{di} \right\} \tag{2}$$

where, ***i=1,2,3,..N***, and $\boldsymbol{x_{11}^{di}}$ as an example, denotes to the first pixel in the first edge $\boldsymbol{E_1}$ of door ***i***. For measuring the intensity distance $\boldsymbol{\gamma(x)}$ between the tested pixel and its surrounding ones, we propose the following formula as:

$$\gamma(x) = e^{\frac{|x-x0|}{I}-1} \quad , \tag{3}$$

where $\boldsymbol{x \subset \{ E_1^i \cup E_2^i \}}$ pixel $\boldsymbol{x}$ is a pixel located in $\boldsymbol{E_1}$or $\boldsymbol{E_2}$ around the tested pixel $\boldsymbol{x_0}$. ***I*** is constant and it is measured by taking the average intensity differences between the pixels in different images. From 3, two vectors $\boldsymbol{\Gamma_1}$ and $\boldsymbol{\Gamma_2}$ are attained as:

$$\Gamma_1^i = \left\{ \gamma_{11}^{di}, \gamma_{21}^{di}, \gamma_{31}^{di}, \ldots\ldots, \gamma_{(5+4(i-1))1}^{di} \right\} \tag{4}$$

$$\Gamma_2^i = \left\{ \gamma_{12}^{di}, \gamma_{22}^{di}, \gamma_{32}^{di}, \ldots\ldots, \gamma_{(5+4i)2}^{di} \right\} \tag{5}$$

where, the best similarity between $\boldsymbol{x_0}$ and $\boldsymbol{x}$ is obtained when $\boldsymbol{x_0=x}$, and the worst similarity is attained when the absolute difference between $\boldsymbol{x_0}$ and $\boldsymbol{x}$ is maximum, i.e., $|\boldsymbol{x- x_0}|$***=255***. One can conclude that the bounds of $\boldsymbol{\gamma(x)}$ are described as:

$$e^{-1} \le \gamma(x) \le \{(255/I)-1\} \tag{6}$$

if $|\boldsymbol{x- x0}| \le \boldsymbol{I}$, then $\boldsymbol{\gamma(x) \le 1}$. Therefore, the range of $\boldsymbol{\gamma(x)}$ that indicates the connectivity or similarity between $\boldsymbol{x_0}$ and $\boldsymbol{x}$ is specified as:

$$e^{-1} \le \gamma(x) \le 1 \tag{7}$$

But if $|x - x0| > 1$, then $\gamma(x) > 1$. That means x_0 is dissimilar to x.

From equations (4), (5), and (7), two vectors A_1^i of size m for E_1 and A_2^i of size n for E_2 are obtained. Each element in the two vectors has similarity value $\gamma \le 1$and in this case, γ is replaced by the parameter a. As an example, if $\gamma_{11}^{d_i} \le 1$ then $\gamma_{11}^{d_i} = a_{11}^{d_i}$

$$A_1^i = \left\{ a_{11}^{di}, a_{21}^{di}, a_{31}^{di}, \ldots\ldots, a_{m)}^{di} \right\} \tag{8}$$

$$A_2^i = \left\{ a_{12}^{di}, a_{22}^{di}, a_{32}^{di}, \ldots\ldots, a_{n}^{di} \right\} \tag{9}$$

The verdict of the originality of the tested pixel is based on the similarity parameter S that is defined by using equations (8) and (9) as:

$$S = \frac{1}{b} \cdot \left[\frac{\left(\sum_{r=1}^{m} a_{r1}^{di} + \sum_{r=1}^{n} a_{r2}^{di} \right)}{n + m} \right] \tag{10}$$

Note that, $b = 1$ if $n + m \ge 2$, otherwise $b<1$. That means, the tested pixel should have similarity at least with other two neighboring pixels as explained in Fig.1. The verdict of the originality of the tested pixel x_0 is expressed as the following:

$$x_0 = \{x_0 \mid S \le 1\} \cup \{x_{no} \mid s > 1\} \tag{11}$$

The noisy pixel x_{no} is restored by taking the median of the four or the eight pixels X that are surrounding the detected noisy pixel at the location ij as:

$$x_{ij}^{rest} = median\ (X_{ij}) \tag{12}$$

3 Simulation Results

The proposed algorithm is tested through several simulation experiments. In which, different corrupted document images are restored. Tested images are corrupted artificially either by fixed valued or random valued impulse noise at different rates. The visual qualities of the restored images reflect the strength of the proposed algorithm and show its ability in preserving the image details. Images of different font sizes are used through the experiments. The advantage of the proposed algorithm is that it can rapidly and efficiently discriminate between the noisy and the original pixel by using several consecutive tests. Any pixel to be considered as original pixel it should be an element in a series of similar pixels. All the experiments are carried out by using MATLAB version 7.2. Figs. 3 and 4 demonstrate the restoration performance of the proposed algorithm in restoring two document images corrupted with random valued impulse noise. In Fig.3 post-office image is corrupted with 10% noise. It is clear that although Fig.3 has lines, picture, handwritten text, the noise is removed efficiently and the image details are preserved. In Fig.4 machine-printed text with 5% random valued impulse noise is used. In Fig.5, hand-written text corrupted with 10% salt and pepper noise is restored. It is clear that the proposed algorithm has removed the noise efficiently and maintains the image details in the different tested images. In Fig.6, the new algorithm is compared with two known methods ACWM filter and TSM filter[16-17] in restoring 5% corrupted document image. It is obvious that the proposed method delivers the best performance since it preserves the fine details and almost removes all the noisy pixels. The proposed algorithm is very fast since it takes times between 5 to 37 seconds to restore the simulated experiments. In all the experiments, three doors are used in the detection process. More extra doors are needed for more heavily corrupted images, since in every door part of the noisy pixels are detected. Thus, we have to increase the number of the doors until the image becomes fully recovered. Parameter ***I*** which represents the maximum difference in intensity between two similar pixels is equal to 40 in the first three figures. In Fig. 5 ***I*** may be increased, the reason is that fixed valued impulse noise has two level of noise either 0 or 255, while random valued impulse noise has 256 levels between [0,255].Thus, in Fig.5 ***I***=80 since the results do not depend precisely on the value of ***I*** and it is easy to find its value or values that deliver the optimum performance.

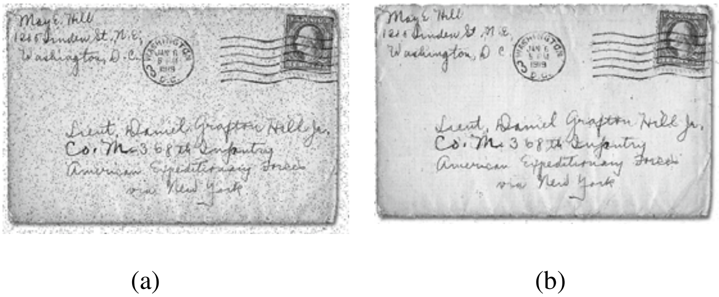
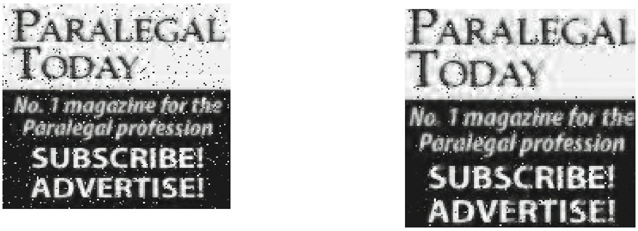

(a) (b)

Fig. 3. Restoration of 10% corrupted post-office document image with random valued impulse noise :(a) Corrupted version (b) Restored version. Consumed time: 36.3 seconds.



PARALEGAL TODAY

No. 1 magazine for the Paralegal profession

SUBSCRIBE! ADVERTISE!

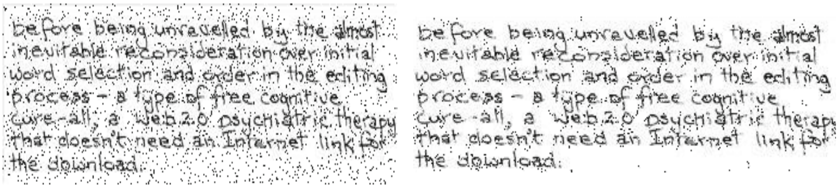
PARALEGAL TODAY

No. 1 magazine for the Paralegal profession

SUBSCRIBE! ADVERTISE!

(a) (b)

Fig. 4. Restoration of 5% corrupted machine-printed document image with random valued impulse noise: (a) Corrupted version (b) Restored version. Consumed time: 5 seconds.

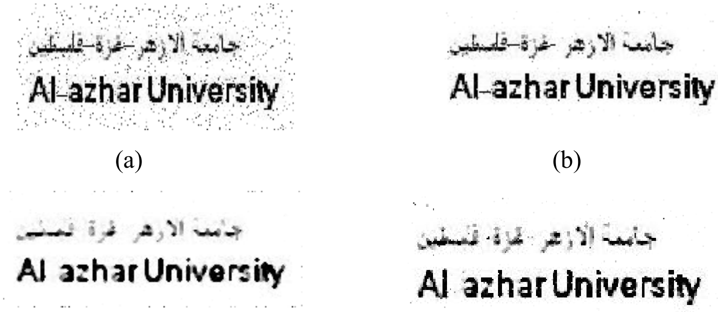


before being unravelled by the almost inevitable reconsideration over initial word selection and order in the editing process – a type of free cognitive cure-all, a web2.0 psychiatric therapy that doesn't need an Internet link for the download.

before being unravelled by the almost inevitable reconsideration over initial word selection and order in the editing process – a type of free cognitive cure-all, a web2.0 psychiatric therapy that doesn't need an Internet link for the download.

(a) (b)

Fig. 5. Restoration of 10%corrupted hand-written document image with salt and pepper impulse noise :(a) Corrupted version (b) Restored version. Consumed time: 12.5 seconds.



جامعة الازهر -غزة- فلسطين

Al-azhar University

جامعة الازهر -غزة- فلسطين

Al-azhar University

(a) (b)

(c) (d)

Fig. 6. Restoration of 5% corrupted machine-printed document image with random valued impulse noise: (a) Corrupted version (b) New method, (c) ACWMF[17], (d) TSMF[18]

4 Conclusion

Corrupted document images are restored in this paper. The restoration process is performed by calculating first norm, degree of similarity, and range of similarity. The current pixel to be considered original pixel, it should fulfill the conditions in different door's sizes. Simulation experiments demonstrate that the proposed method restores efficiently and rapidly the corrupted documents by salt and pepper noise, or random valued impulse noise. The restored images are readable and maintain the image details.

References

1. Meloche, J., Zamar, R.H.: Binary Image Restoration. Canadian J. Stat. 22, 335–355 (1994)
2. Hitchcock, D., Glasbey, C.A.: Binary Image Restoration at Sub-Pixel Resolution. Biomet. 53, 1010–1053 (1997)
3. Heanue, J., Gurkan, K., Hesselink, L.: Signal Detection for Page-Access Optical Memories with Intersymbol Interference. Appl. Optics (1996)
4. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear Total Variation Based Noise Removal Algorithms. Phys. D 60, 259–268 (1992)
5. Chan, T., Esedoglu, S., Nikolova, M.: Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models. Tech. Rep. CAM report 04-54 (2004)
6. Aubert, G., Vese, L.: A Variational Method in Image Recovery. SIAM J. Numer. Anal. 34, 1948–1979 (1997)
7. Chan, T.F., Esedoglu, S., Nikolova, M.: Finding the Global Minimum for Binary Image Restoration. In: ICIP 2005, pp. 121–124 (2005)
8. Kapur, J.N., et al.: A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram. Comput. Vision Graph: Image Proc. 29, 273–285 (1985)
9. Wang, S., Haralick, R.: Automatic Threshold Selection Computer Vision Graph. Image Proc. 25, 46–67 (1984)
10. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. IEEE Trans. Syst. Man Cybern. 9, 62–66 (1979)
11. Leedham, G., Yan, C., Takru, K., Tan, J.H.N., Mian, L.: Comparison of Some Thresholding Algorithms for Text/Background Segmentation in Difficult Document Image. In: Proc. 7th ICDAR. IEEE Press, New York (2003)
12. Mitra, A.: Restoration of Noisy Document Images with an Efficient Bi-level Adaptive Thresholding. World Academy Sci. Engin. Tech. 18, 216–221 (2006)
13. Zhang, Y., Wu, L.: A Fast Document Image Denoising Method Based on Packed Binary Format and Source Word Accumulation. J. Converg. Inform. Tech. 6, 131–137 (2011)
14. Bouaynaya, N., Charif-Chefchaouni, M., Schon-feld, D.: Theoretical Foundations of Spatially-Variant Mathematical Morphology Part I: Binary Images. IEEE Trans. Pattern Anal. Mach. Intell. 30, 823–836 (2008)
15. Chatzis, V.: A Generalized Fuzzy Mathematical Morphology and Its Application in Robust 2-D and 3-D Object Representation. IEEE Trans. Image Process. 9, 1798–1810 (2000)
16. Schonfeld, D.: Optimal Strcuturing Elements for the Morphological Pattern Restoration of Binary Images. IEEE Trans. Pattern Anal. Mach. Intell. 16, 589–601 (1994)
17. Chen, T., Wu, H.R.: Adaptive Impulse Detection Using Center-Weighted Median Filters. IEEE Signal Process. Lett. 8, 1–3 (2001)
18. Chen, T., Ma, K.K., Chen, L.H.: Tri-State Median Filter for Image Denoising. IEEE Trans. Image Process. 8, 1834–1838 (1999)

Evolving Flexible Beta Operator Neural Trees (FBONT) for Time Series Forecasting

Souhir Bouaziz*, Habib Dhahri, and Adel M. Alimi

REsearch Group on Intelligent Machines (REGIM), University of Sfax, National School of Engineers (ENIS), BP 1173, Sfax 3038, Tunisia
souhir.bouaziz@gmail.com, {habib.dhahri,adel.alimi}@ieee.org

Abstract. In this paper, a new time-series forecasting model based on the Flexible Beta Operator Neural Tree (FBONT) is introduced. The FBONT model which has a tree-structural representation is considered as a special Beta basis function multi-layer neural network. Based on the pre-defined Beta operator sets, the FBONT can be formed and optimized. The FBONT structure is developed using the Extended Genetic Programming (EGP) and the Beta parameters and connected weights are optimized by the Particle Swarm Optimization algorithm (PSO). The performance of the proposed method is evaluated using time series forecasting problems and compared with those of related methods.

Keywords: Flexible Beta Operator Neural Tree model, Extended Genetic Programming, Particle Swarm Optimization algorithm, Time-series forecasting.

1 Introduction

Time series forecasting play a major role in the characterization of time series performance by predicting the future value and understanding fundamental features in systems, so it has been a center of attention of several researches. Recently, various nonlinear time series forecasting methods have been proposed such as artificial neural networks (ANN) [1, 2, 3, 4], SVM [5], adaptive algorithms [6, 7], and have been successfully applied.

A neural network's performance depends mainly on two issues which are the network structure and the parameter's adjustment on the continuous parameter space and these issues are closely coupled. For a given problem, the neural network structure is not unique and also it may be a single hidden layer is not enough. Thus, the design of ANN automatically is required and many important attempts have been developed such as evolutionary programming [8], Neuro Evolution of augmenting topologies [9]. Furthermore, weights and kernel parameters of ANNs can be learned by many methods, i.e., back-propagation algorithm [10], genetic algorithm [11], differential evolution algorithm [1, 3, 4], particle swarm optimization algorithm [2] and so on.

Although conventional representation of ANN has the nonlinear approximation capability, it also presents many weaknesses, for example, the neural network's structure

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 17–24, 2012.

is difficult to regulate, it suffers from slow convergence characteristics and over-fitting phenomenon leading the decline of its generalization, it is prone to be trapped in local minima [12]. Thus a special multi-layer feedforward ANN has been proposed by Chen [13] and it is called flexible neural tree (FNT). FNT allows over-layer connections, input variables selection and different activation functions for different nodes [12]. Recent studies have begun to explore this representation of neural networks in the context of classification [14], recognition [15], approximation [16] and control [17], etc. To form the flexible neuron model, the most used flexible activation function is the Gaussian function. However, the Beta function [18, 19] shows its performance for standard representation of ANN against the Gaussian function due to its great flexibility and its universal approximation ability [1-4, 11]. For these reasons we adopted in this research the flexible Beta function to establish the flexible neuron model.

In this paper, a flexible Beta operator neural tree (FBONT) model is proposed for time-series prediction problem. Based on the predefined Beta operator sets, a flexible Beta operator neural tree model can be created and evolved. The hierarchical structure is evolved using the Extended Genetic Programming (EGP). The fine tuning of the Beta parameters (centre, spread and the form parameters) and weights encoded in the structure is accomplished using the Particle Swarm Optimization algorithm (PSO).

The paper is planned as follows: Section 2 describes the basic flexible Beta operator neural tree model. A hybrid learning algorithm for evolving the Beta function neural tree models is the subject of Section 3. The set of some simulation results are provided in Section 4. Finally, some concluding remarks are presented in Section 5.

2 Flexible Beta Operator Neural Tree Model

In this work, we have used the tree-based encoding method as it defined by Chen [12-17] for representing a FBONT model. The function node set F and terminal node set T used for generating a FBONT model are described as follows:

$$S = F \cup T = \{+_2, +_3, \dots, +_N\} \cup \{x_1, \dots, x_M\} \tag{1}$$

where $+_n$ (n = 2,. . . , N) denote non-terminal nodes and represent flexible neuron Beta operators with n inputs.

$x_1, x_2, . . ., x_M$ are terminal nodes and defining the input vector values. The output of a non-terminal node is calculated as a flexible neuron model (fig.1).

In the creation process of Beta operator neural tree, if a non-leaf node, i.e., $+_n$ is selected, n real values are randomly created to represent the connected weight between the node $+_n$ and its offspring. In addition, seen that the flexible activation function used in this study is the beta function, four adjustable parameters (the center c_n, width σ_n and the form parameters p_n, q_n) are randomly generated as flexible Beta operator parameters. For each non-terminal node, i.e., $+_n$, its total excitation is calculated by:

$$y_n = \sum_{j=1}^{n} w_j * x_j \tag{2}$$

where x_j(j = 1, …, n) are the inputs to node $+_n$. The output of node $+_n$ is then calculated by:

$$out_n = \beta(y_n, c_n, \sigma_n, p_n, q_n) = \begin{cases} \left[1 + \frac{(p_n + q_n)(y_n - c_{\textcircled{T}})}{\sigma_n p_n}\right]^{p_n} \left[1 - \frac{(p_n + q_n)(c_n - y_n)}{\sigma_n q_n}\right] q_n \\ \qquad\qquad if\ y_n \in \left]c_n - \frac{\sigma_n p_n}{p_n + q_n}, c_n + \frac{\sigma_n q_n}{p_n + q_n}\right[\\ 0 \qquad\qquad else \end{cases} \tag{3}$$

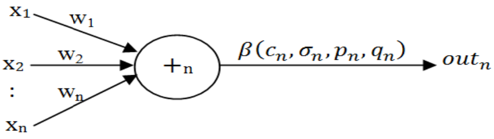


Fig. 1. A flexible neuron Beta operator

A typical flexible Beta operator neural tree model is shown as Fig. 2. The overall output of flexible Beta operator neural tree can be computed recursively by depth-first method from left to right.

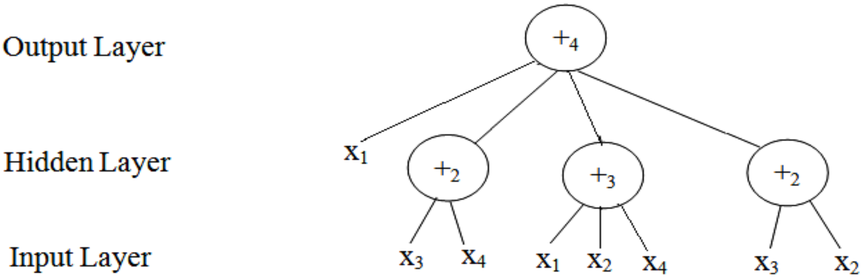


Fig. 2. A typical representation of FBONT: function node set F = {$+_2$, $+_3$, $+_4$}, and terminal node set T = {x_1, x_2, x_3, x_4}

3 The Hybrid FBONT Evolving Algorithm

The optimization of FBONT includes the tree-structure and parameter optimization. In this study, finding an optimal Beta operator neural tree structure is achieved by using Extended Genetic Programming algorithm and the parameters implanted in a FBONT are optimized by PSO.

3.1 Structure Optimization

A number of flexible neural tree variation operators, which are an extension of standard GP, are developed in this work as following:

Selection: In this study firstly a truncation selection is used by ranking all individuals according to their fitness. Then, a threshold *T* (between 0 and 1) is applied such that the *(1-T)%* best individuals are selected to survive to the next generation and the remaining individuals are removed and replaced with new ones.

Crossover: the tree structure crossover operation is implemented by taking randomly selected sub-trees in the individuals and selecting randomly one non-leaf node in the hidden layer for each chromosome, and then swapping the selected sub-trees.

Mutation: four different mutation operators were used to generate offspring from the parents. These mutation operators are as follows:

1. *Changing one leaf node*: select one leaf node randomly in the neural Beta operator tree and replace it with another leaf node;
2. *Changing all the leaf nodes:* select all leaf nodes in the neural Beta operator tree and replace it with another leaf node;
3. *Growing:* select a random leaf node in hidden layer of the neural Beta operator tree and replace it with a randomly generated sub-tree;
4. *Pruning:* randomly select a Beta operator node in the neural tree and replace it with a random leaf node.

After each mutation or crossover operator, a redundant terminals pruning operator will be applied, if it is possible; i.e. if a Beta operator node has more than two terminals, the redundant terminals should be deleted.

3.2 Parameter Optimization with PSO

PSO was proposed by Kennedy and Eberhart [20] and is inspired by the swarming behavior of animals. The initial population of particles is randomly generated. Each particle has a position vector denoted by x_i. A swarm of particles 'flies' through the search space; with the velocity vector v_i of each particle. Each particle records its best position corresponding to the best fitness in a vector p_i. Moreover, the best position among all the particles obtained in a certain neighborhood of a particle is recorded in a vector p_g. At each iteration, a new velocity for particle *i* is updated by:

$$v_i(t+1) = w\, v_i(t) + c_1\varphi_1\big(p_i(t) - x_i(t)\big) + c_2\varphi_2\left(p_g(t) - x_i(t)\right) \tag{4}$$

where c_1,c_2 (acceleration) and w (inertia) are positive constant and φ_1 and φ_2 are randomly distributed number in [0,1]. The velocity v_i is limited in[-v_{max} ,+v_{max}]. Based on the calculated velocities, each particle changes its position:

$$x_i(t+1) = x_i(t) + (1-w)v_i(t+1) \tag{5}$$

3.3 Fitness Function

To find an optimal FBONT, the Root Mean Squared Error (RMSE) is employed as a fitness function:

$$Fit(i) = \sqrt{\frac{1}{P}\sum_{j=1}^{P}(y_t^j - y_{out}^j)^2} \tag{6}$$

where P is the total number of samples, y_t^j and y_{out}^j are the actual time-series and the FBONT model output of jth sample. $Fit(i)$ denotes the fitness value of ith individual.

3.4 The Learning Algorithm for FBONT Model

To find an optimal or near-optimal FBONT model, architecture and parameters optimization are used alternately. Combining of the EGP and PSO algorithms, a hybrid algorithm for evolving FBONT model is described as follows and is depicted:

(a) Randomly create an initial population (FBONT trees and its parameters);
(b) Structure optimization is achieved by the Extended Genetic Programming (EGP) as described in section 3.1;
(c) If a better architecture is found or a maximum number of generation is attained, then go to step (d), otherwise go to step (b);
(d) Parameter optimization is achieved by the PSO algorithm. The architecture of FBONT model is fixed, and it is the best tree found by the structure search. The parameters (weights and flexible Beta function parameters) encoded in the best tree formulate a particle;
(e) If the maximum number of iterations is attained, or no better parameter vector is found for a fixed time then go to step (f); otherwise go to step (d);
(f) If satisfactory solution is found, then the algorithm is stopped; otherwise go to step (b).

4 Experimental Results

To evaluate its performance, the proposed FBONT model is submitted to time-series prediction problems: Mackey-Glass chaotic and the Jenkins–Box time series.

4.1 Mackey–Glass Time Series Prediction

A time-series prediction problem can be constructed based on the Mackey–Glass [21] differential equation:

$$\frac{d(x(t))}{dt} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t) \tag{7}$$

The setting of the experiment varies from one work to another. In this work, the same parameters of [2] and [14], namely a = 0.2, b = 0.1 and $\tau \geq 17$, were adopted, since the results from these works will be used for comparison. 500 data pairs of the series were used as training data, and 500 were used to validate the model identified. The used Beta operator sets to create an optimal FBONT model is $S = F \cup T = \{+_2, +_3, +_4, +_5\} \cup \{x_1, x_2, x_3, x_4\}$, where x_i (i = 1, 2, 3, 4) denotes $x(t), x(t -$

$6), x(t-12)$ and $x(t-18)$, respectively. After 16 generations, an optimal FBONT model was obtained with RMSE 0.007401. The RMSE value for validation data set is 0.007623. The evolved FBONT and its output are shown in Fig. 3. The proposed system is essentially compared with beta basis function neural network's system using particle swarm optimization (BBFN-PSO) [2] and the FNT model with Gaussian function as flexible neuron operator [13] and also with other systems in Table 1.

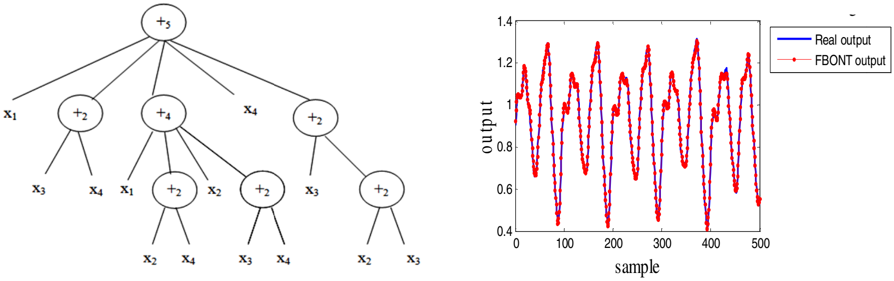


Fig. 3. The evolved FBONT and its output for prediction of the Mackey-Glass time-series

Table 1. Comparison of different methods for the Mackey-Glass time-series

Method	**Prediction error (RMSE)**
PSO-BBFN [2]	0.027
Habib [4]	0.013
Aouiti [11]	0.013
FNT [13]	0.0069
FBONT	0.0076

4.2 Box and Jenkins' Gas Furnace Problem

The gas furnace data of Box and Jenkins [22] was saved from a combustion process of a methane–air mixture. It is frequently used as a benchmark example for testing prediction algorithms. The input $u(t)$ is the gas flow into the furnace and the output $y(t)$ is the CO_2 concentration in outlet gas. In this work, 200 data samples are used for training and 96 data samples are used for testing the performance of the evolved model. The used instruction set for creating a FBONT model $S = F \cup T = \{+_2, +_3, +_4\} \cup \{_1, x_2\}$, where x_i (i = 1, 2) denotes $u(t-4), y(t-1)$, respectively. After 16 generations, the optimal Beta operator neural tree model was obtained with the MSE 0.000023. The MSE value for validation data set is 0. 0.000135. The evolved FBONT and its output are shown in Fig. 4. A comparison result of different methods for forecasting Jenkins–Box data is shown in Table 2.

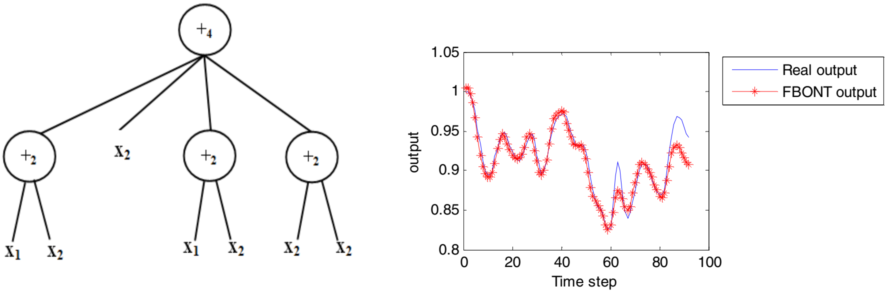


Fig. 4. The evolved FBONT and its output for forecasting Jenkins–Box data

Table 2. Comparison of testing errors of Box and Jenkins

Method	**Prediction error (MSE)**
ANFIS model [23]	0.0073
FuNN model [24]	0.0051
FNT [14]	0.00066
HMDDE [3]	0.0581
FBONT	**0.000135**

5 Conclusion

In this paper, a Flexible Beta Operator Neural Tree model and its design and optimization algorithm are proposed for time-series forecasting problems. The work demonstrates that the FBONT model can successfully evolve the structure and parameters of artificial neural networks simultaneously by using a tree representation. In fact, the FBONT structure is developed using Extended Genetic Programming (EGP) and the Beta parameters and connected weights are optimized by Particle Swarm Optimization algorithm (PSO). The experiment results show that the FBONT model can effectively predict the time-series problem such as Mackey-Glass chaotic time series and the Jenkins–Box time series.

Acknowledgments. The authors would like to acknowledge the financial support of this work by grants from General Direction of Scientific Research (DGRST), Tunisia, under the ARUB program.

References

1. Dhahri, H., Alimi, A.M.: Opposition-based Differential Evolution for Beta Basis Function Neural Network. In: IEEE Congress on Evolutionary Computation, Barcelona, Spain, pp. 1–8 (2010)
2. Dhahri, H., Alimi, A.M., Karray, F.: Designing Beta Basis Function Neural Network for Optimization Using Particle Swarm Optimization. In: IEEE International Joint Conference on Neural Networks, Hong Kong, China, pp. 2564–2571 (2008)

3. Dhahri, H., Alimi, A.M., Abraham, A.: Hierarchical multi-dimensional differential evolution for the design of beta basis function neural network. Neurocomputing 79, 131–140 (2012)
4. Dhahri, H., Alimi, A.M.: The Modified Differential Evolution and the RBF (MDE-RBF) Neural Network for Time Series Prediction. In: Proc. of the International Conference, pp. 5245–5250 (2006)
5. Liu, H., Liu, D., Deng, L.-F.: Chaotic Time Series Prediction Using Fuzzy Sigmoid Kernel-based Support Vector Machines. Chin. Phys. 15(6), 1196–1200 (2006)
6. Li, H., Zhang, J., Xiao, X.: Neural Volterra Filter for Chaotic Time Series Prediction. Chin. Phys. 14(11), 2181–2188 (2005)
7. Meng, Q., Zhang, Q., Mu, W.: A Novel Multi-step Adaptive Prediction Method for Chaotic Time Series. Acta Phys. Sin. 55(4), 1666–1671 (2006)
8. Yao, X., Liu, Y., Lin, G.: Evolutionary Programming Made Faster. IEEE Trans. Evolut. Comput. 3, 82–102 (1999)
9. Stanley, K.O., Miikkulainen, R.: Evolving Neural Networks through Augmenting Topologies. Evolut. Comput. 10, 99–127 (2002)
10. Stepniewski, S.W., Keane, A.J.: Pruning Back-propagation Neural Networks Using Modern Stochastic Optimization Techniques. Neural Comput. Appl. 5, 76–98 (1997)
11. Aouiti, C., Alimi, A.M., Maalej, A.: A Genetic Designed Beta Basis Function Neural Networks for Approximating of Multi-variables Functions. In: Proc. Int. Conf. Artificial Neural Nets and Genetic Algorithms. Springer Computer Science, Prague, Czech Republic, pp. 383–386 (2001)
12. Chen, Y., Yang, B., Meng, Q.: Small-time Scale Network Traffic Prediction Based on Flexible Neural Tree. Appl. Soft Comput. 12, 274–279 (2012)
13. Chen, Y., Yang, B., Dong, J., Abraham, A.: Time-series Forecasting Using Flexible Neural Tree Model. Inf. Sci. 174, 219–235 (2005)
14. Chen, Y., Abraham, A., Yang, B.: Feature Selection and Classification using Flexible Neural Tree. Neurocomput. 70, 305–313 (2006)
15. Chen, Y., Jiang, S., Abraham, A.: Face Recognition Using DCT and Hybrid Flexible Tree. In: Proc. of the International Conference on Neural Networks and Brain, pp. 1459–1463 (2005)
16. Chen, Y., Peng, L., Abraham, A.: Exchange Rate Forecasting Using Flexible Neural Trees. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3973, pp. 518–523. Springer, Heidelberg (2006)
17. Chen, Y., Meng, Q., Zhang, Y.: Optimal Design of Hierarchical B-Spline Networks for Nonlinear System Identification. J. Dynam. Continu. Discrete Impul. Systems Series B (2006)
18. Alimi, A.M.: The Beta System: Toward a Change in Our Use of Neuro-Fuzzy Systems. Int. J. Manag. Invited Paper, 15–19 (2000)
19. Alimi, A.M.: The Beta Fuzzy System: Approximation of Standard Membership Functions. In: Proc. 17eme Journees Tunisiennes d'Electrotechnique et d'Automatique: JTEA 1997, Nabeul, Tunisia, pp. 108–112 (1997)
20. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Press, Piscataway (1995)
21. Lzvbjerg, M., Krink, T.: Extending particle swarms with self-organized criticality. In: Proceedings of the Fourth Congress on Evolutionary Computation (CEC 2002), pp. 1588–1593. IEEE Press, Piscataway (2002)
22. Box, G.E.P., Jenkins, G.M.: Time Series Analysis-Forecasting and Control. Holden Day, San Francisco (1976)
23. Nie, J.: Constructing fuzzy Model by Self-organising Counter Propagation Network. IEEE Trans. Systems Man Cybern. 25, 963–970 (1995)
24. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-fuzzy and Soft Computing: a Computational Approach to Learning and Machine Intelligence. Prentice-Hall, Upper Saddle River (1997)

Nonparametric Localized Feature Selection via a Dirichlet Process Mixture of Generalized Dirichlet Distributions

Wentao Fan* and Nizar Bouguila

Concordia Institute for Information Systems Engineering
Concordia University, QC, Canada
wenta_fa@encs.concordia.ca, nizar.bouguila@concordia.ca

Abstract. In this paper, we propose a novel Bayesian nonparametric statistical approach of simultaneous clustering and localized feature selection for unsupervised learning. The proposed model is based on a mixture of Dirichlet processes with generalized Dirichlet (GD) distributions, which can also be seen as an infinite GD mixture model. Due to the nature of Bayesian nonparametric approach, the problems of overfitting and underfitting are prevented. Moreover, the determination of the number of clusters is sidestepped by assuming an infinite number of clusters. In our approach, the model parameters and the local feature saliency are estimated simultaneously by variational inference. We report experimental results of applying our model to two challenging clustering problems involving web pages and tissue samples which contain gene expressions.

Keywords: Mixture Models, Clustering, Dirichlet Process, Nonparametric Bayesian, Generalized Dirichlet, Localized Feature Selection, Variational Inference.

1 Introduction

Clustering is a critical technique in unsupervised learning problems which is used to partition the data into homogeneous groups. While there exist many algorithms for clustering, we focus on finite mixture models, which is one of the most powerful techniques and has been widely applied in many fields such as data mining, machine learning, image processing and bioinformatics. Among various mixture models, Gaussian mixture model has been a popular choice due to its simplicity and maturity of relevant techniques [13,16]. However, when the data clearly appears with a non-Gaussian structure, other distributions may provide better modeling capabilities, such as the generalized Dirichlet (GD) distributions [6,7]. One of the most challenging issues regarding finite mixture models is to determine the appropriate number of clusters underlying the data. According to recent development, a nonparametric Bayesian technique namely Dirichlet process (DP) [12] may provide an elegant solutions to this model selection problem.

* Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 25–33, 2012.

The DP mixture model can be considered as an infinite mixture model, such that its complexity increases as the data set grows. Thanks to the development of Markov chain Monte Carlo (MCMC) techniques, the use of Dirichlet process has been spread across many domains [20,22]. However, in practice, the use of MCMC techniques is highly computational demanding and is often limited to small-scale problems. An alternative to the MCMC technique is a deterministic approximation technique known as variational inference [15,2]. It has received significant attention and has provided promising performance in many applications, especially in finite mixture models [3,10,11,17]. Furthermore, variational inference only requires a modest amount of computational power which makes it suitable to large applications. In [5], the authors have proposed a general variational inference algorithm for DP mixtures with exponential family based on the stick-breaking representation [21]. In their work, the model is a full Dirichlet process and the approximated variational distributions are truncated to yield a finite dimensional representation.

The main purpose of this paper is to develop a novel unsupervised clustering approach based on a nonparametric Bayesian model with variational framework. Our contributions are listed as the following: First, we develop an infinite GD mixture model using stick-breaking construction such that the difficulty of choosing the correct number of components is avoided. Second, rather than using the global (i.e produce a common feature subset for all the mixture components) unsupervised feature selection method which is commonly used in many works [16,9,8], we adopt a localized feature selection scheme [17] where different feature subsets are associated to the different mixture components. The motivation of this particular choice is based on the fact that it has been shown that global feature selection may not be realistic in real life applications and that localized feature selection can generally provide better results [10,17]. Third, we develop a variational inference framework for learning the proposed model, such that the model parameters and the local feature saliency are estimated simultaneously.

The rest of this paper is organized as follows: In Section 2, we develop the infinite GD mixture model with localized feature selection scheme and learn it through the variational inference. Section 3 is devoted to the experimental results. Finally, conclusion is provided in Section 4.

2 Model Specification and Variational Learning

Given a random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_D)$ which is drawn from a finite mixture of generalized Dirichlet (GD) Distributions with M components, such that [7]: $p(\boldsymbol{Y}|\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{j=1}^{M} \pi_j \mathrm{GD}(\boldsymbol{Y}|\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j)$, where $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M\}$ and $\boldsymbol{\beta} = \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_M\}$. $\boldsymbol{\alpha}_j$ and $\boldsymbol{\beta}_j$ are the parameters of the GD distribution representing component j with $\boldsymbol{\alpha}_j = \{\alpha_{j1}, \ldots, \alpha_{jD}\}$ and $\boldsymbol{\beta}_j = \{\beta_{j1}, \ldots, \beta_{jD}\}$. $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_M\}$ denotes the mixing coefficients, subject to the constraints: $0 \leq \pi_j \leq 1$, $\sum_{j=1}^{M} \pi_j = 1$. The GD distribution of $\boldsymbol{Y}$ with parameters $\boldsymbol{\alpha}_j$ and $\boldsymbol{\beta}_j$ is defined as

$$\mathrm{GD}(\boldsymbol{Y}|\boldsymbol{\alpha}_j,\boldsymbol{\beta}_j)=\prod_{d=1}^{D}\frac{\Gamma(\alpha_{jd}+\beta_{jd})}{\Gamma(\alpha_{jd})\Gamma(\beta_{jd})}Y_d^{\alpha_{jd}-1}\Bigg(1-\sum_{k=1}^{d}Y_k\Bigg)^{\gamma_{jd}} \tag{1}$$

where $\sum_{d=1}^{D}Y_d<1$ and $0<Y_d<1$ for $d=1,\ldots,D$, $\alpha_{jd}>0$, $\beta_{jd}>0$, $\gamma_{jd}=\beta_{jd}-\alpha_{jd+1}-\beta_{jd+1}$ for $d=1,\ldots,D-1$, and $\gamma_{jD}=\beta_{jD}-1$. In this paper, following an interesting mathematical property of the GD distribution which is thoroughly discussed in [8], we can rewrite the finite GD mixture model in the following form

$$p(\boldsymbol{X}_i|\boldsymbol{\pi},\boldsymbol{\alpha},\boldsymbol{\beta})=\sum_{j=1}^{M}\pi_j\prod_{d=1}^{D}\mathrm{Beta}(X_{id}|\alpha_{jd},\beta_{jd}) \tag{2}$$

where $\boldsymbol{X}_i=(X_{i1},\ldots,X_{iD})$, $X_{i1}=Y_{i1}$ and $X_{id}=Y_{id}/(1-\sum_{k=1}^{d-1}Y_{ik})$ for $d>1$, and $\mathrm{Beta}(X_{id}|\alpha_{jd},\beta_{jd})$ is a Beta distribution defined with parameters (α_{jd},β_{jd}). The motivation of adopting this property is that the independence between the features now becomes a fact rather than an assumption as considered in previous unsupervised feature selection Gaussian mixture-based approaches [16,9].

In this paper, we construct our Dirichlet process using the stick-breaking construction. Specifically, given a random distribution G, it is distributed according to a Dirichlet process ($G\sim DP(\psi,H)$) if the following conditions are satisfied:

$$\lambda_j\sim\mathrm{Beta}(1,\psi),\qquad\theta_j\sim H,\qquad\pi_j=\lambda_j\prod_{s=1}^{j-1}(1-\lambda_s),\qquad G=\sum_{j=1}^{\infty}\pi_j\delta_{\theta_j} \tag{3}$$

where δ_{θ_j} denotes the Dirac delta measure centered at θ_j. The mixing weights π_j are obtained by recursively breaking a unit length stick into an infinite number of pieces such that the size of each successive piece is proportional to the rest of the stick. We then extent (2) into an infinite mixture model by assuming that the observed data set is generated from a GD mixture model with a countably infinite number of components as: $p(\boldsymbol{X}_i|\boldsymbol{\pi},\boldsymbol{\alpha},\boldsymbol{\beta})=\sum_{j=1}^{\infty}\pi_j\prod_{d=1}^{D}\mathrm{Beta}(X_{id}|\alpha_{jd},\beta_{jd})$.

Next, we deploy a localized feature selection scheme [17] which has been shown to outperform the global one. Thus, the distribution of each feature X_{id} can be approximated by $p(X_{id})\simeq\mathrm{Beta}(X_{id}|\alpha_{jd},\beta_{jd})^{\phi_{ijd}}\mathrm{Beta}(X_{id}|\sigma_{jd},\tau_{jd})^{1-\phi_{ijd}}$, where ϕ_{ijd} is a binary latent variable and known as the feature relevance indicator, such that $\phi_{ijd}=0$ if feature d of component j is irrelevant (i.e. noise) and follows a Beta distribution: $\mathrm{Beta}(X_{id}|\sigma_{jd},\tau_{jd})$. The prior distribution of $\boldsymbol{\phi}$ is defined as: $p(\boldsymbol{\phi}|\boldsymbol{\epsilon})=\prod_{i=1}^{N}\prod_{j=1}^{\infty}\prod_{d=1}^{D}\epsilon_{jd_1}^{\phi_{ijd}}\epsilon_{jd_2}^{1-\phi_{ijd}}$, where each ϕ_{ijd} is a Bernoulli variable such that $p(\phi_{ijd}=1)=\epsilon_{jd_1}$ and $p(\phi_{ijd}=0)=\epsilon_{jd_2}$. The vector $\boldsymbol{\epsilon}$ represents the features saliencies (i.e. the probabilities that the features are relevant) where $\boldsymbol{\epsilon}_{jd}=(\epsilon_{jd_1},\epsilon_{jd_2})$ and $\epsilon_{jd_1}+\epsilon_{jd_2}=1$. In addition, a Dirichlet distribution is placed over $\boldsymbol{\epsilon}$ with positive parameter $\boldsymbol{c}$: $p(\boldsymbol{\epsilon})=\prod_{j=1}^{\infty}\prod_{d=1}^{D}\mathrm{Dir}(\boldsymbol{\epsilon}_{jd}|\boldsymbol{c})$. Next, a binary latent variable $\boldsymbol{Z}_i=(Z_{i1},Z_{i2},\ldots)$ is placed over each vector $\boldsymbol{X}_i$, such that $Z_{ij}\in\{0,1\}$ and $Z_{ij}=1$ if $\boldsymbol{X}_i$ belongs to component j and 0, otherwise. Thus, by setting $\Omega=\{\mathcal{Z},\boldsymbol{\phi},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\sigma},\boldsymbol{\tau}\}$, we can write the likelihood function of the infinite GD mixtures with latent variables $\mathcal{Z}=(\boldsymbol{Z}_1,\ldots,\boldsymbol{Z}_N)$ as

$$p(\mathcal{X}|\Omega) = \prod_{i=1}^{N}\prod_{j=1}^{\infty}\left[\prod_{d=1}^{D}\mathrm{Beta}(X_{id}|\alpha_{jd},\beta_{jd})^{\phi_{ijd}}\mathrm{Beta}(X_{id}|\sigma_{jd},\tau_{jd})^{1-\phi_{ijd}}\right]^{Z_{ij}} \quad (4)$$

The prior distribution of latent variables $\{Z_{ij}\}$ are given discrete by: $p(\mathcal{Z}|\boldsymbol{\pi}) = \prod_{i=1}^{N}\prod_{j=1}^{\infty}\pi_j^{Z_{ij}}$. According to the stick-breaking construction of DP as stated in (3), $\boldsymbol{\pi}$ is a function of $\boldsymbol{\lambda}$ and we can redefine the probability distribution of $\mathcal{Z}$ as: $p(\mathcal{Z}|\boldsymbol{\lambda}) = \prod_{i=1}^{N}\prod_{j=1}^{\infty}\left[\lambda_j\prod_{s=1}^{j-1}(1-\lambda_s)\right]^{Z_{ij}}$. As shown in (3), $\boldsymbol{\lambda}$ follows a specific Beta distribution: $p(\boldsymbol{\lambda}) = \prod_{j=1}^{\infty}\mathrm{Beta}(1,\psi_j)$. Based on the fact that Gamma distribution is conjugate to the stick lengths of the Dirichlet process mixture model [5], we add another layer to the Bayesian hierarchy for the sake of more flexibility by placing a Gamma prior $\mathcal{G}(\cdot)$ over the hyperparamete $\boldsymbol{\psi}$: $p(\boldsymbol{\psi}) = \mathcal{G}(\boldsymbol{\psi}|\boldsymbol{a},\boldsymbol{b})$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are positive. Last, we need to introduce conjugate priors over parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ of Beta distributions. Here, as proposed in [18], we assume that these Beta parameters are statistically independent and Gamma priors are adopted to approximate the conjugate priors. Thus, we have: $p(\boldsymbol{\alpha}) = \mathcal{G}(\boldsymbol{\alpha}|\boldsymbol{u},\boldsymbol{v})$, $p(\boldsymbol{\beta}) = \mathcal{G}(\boldsymbol{\beta}|\boldsymbol{p},\boldsymbol{q})$, $p(\boldsymbol{\sigma}) = \mathcal{G}(\boldsymbol{\sigma}|\boldsymbol{g},\boldsymbol{h})$ and $p(\boldsymbol{\tau}) = \mathcal{G}(\boldsymbol{\tau}|\boldsymbol{s},\boldsymbol{t})$.

Next, we develop a variational framework for learning the infinite GD mixture model with localized feature selection. To simplify the notation, we define $\Theta = \{\mathcal{Z},\boldsymbol{\phi},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\sigma},\boldsymbol{\tau},\boldsymbol{\lambda},\boldsymbol{\psi},\boldsymbol{\epsilon}\}$. The main idea in variational learning is to find an approximation $Q(\Theta)$ for the posterior distribution $p(\Theta|\mathcal{X})$. Here, we adopt a factorial approximation for the variational inference. Furthermore, motivated by [5], we truncate the stick-breaking representation for the infinite GD mixture model at a value of M as: $\lambda_M = 1$,$\pi_j = 0$ when $j > M$,$\sum_{j=1}^{M}\pi_j = 1$. Here, the truncation level M is a variational parameter which can be freely initialized and will be optimized automatically during the learning process. In variational inference, the general expression for updating a variational factor is given by: $Q_s(\Theta_s) = \big(\exp\langle\ln p(\mathcal{X},\Theta)\rangle_{\neq s}\big)/\big(\int\exp\langle\ln p(\mathcal{X},\Theta)\rangle_{\neq s}\mathrm{d}\Theta\big)$, where $\langle\cdot\rangle_{\neq s}$ denotes an expectation with respect to all the factor distributions except for s. Then, we obtain their variational solutions as

$$Q(\mathcal{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{M}r_{ij}^{Z_{ij}}\ , \quad Q(\boldsymbol{\lambda}) = \prod_{j=1}^{M}\mathrm{Beta}(\lambda_j|\theta_j,\vartheta_j), \quad Q(\boldsymbol{\psi}) = \prod_{j=1}^{M}\mathcal{G}(\psi_j|a_j^*,b_j^*) \quad (5)$$

$$Q(\boldsymbol{\epsilon}) = \prod_{j=1}^{M}\prod_{d=1}^{D}\mathrm{Dir}(\boldsymbol{\epsilon}_{jd}|\boldsymbol{c}^*)\ , \qquad Q(\boldsymbol{\phi}) = \prod_{i=1}^{N}\prod_{j=1}^{M}\prod_{d=1}^{D}f_{ijd}^{\phi_{ijd}}(1-f_{ijd})^{(1-\phi_{ijd})} \quad (6)$$

$$Q(\boldsymbol{\alpha}) = \prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{G}(\alpha_{jd}|u_{jd}^*,v_{jd}^*)\ , \quad Q(\boldsymbol{\beta}) = \prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{G}(\beta_{jd}|p_{jd}^*,q_{jd}^*) \quad (7)$$

$$Q(\boldsymbol{\sigma}) = \prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{G}(\sigma_{jd}|g_{jd}^*,h_{jd}^*)\ , \qquad Q(\boldsymbol{\tau}) = \prod_{j=1}^{M}\prod_{d=1}^{D}\mathcal{G}(\tau_{jd}|s_{jd}^*,t_{jd}^*) \quad (8)$$

where the variational parameters in the above equations are defined as

$$
\begin{aligned}
&r_{ij} = \frac{\exp\left[\sum_{d=1}^{D}\langle\phi_{ijd}\rangle\xi+\sum_{d=1}^{D}\langle 1-\phi_{ijd}\rangle\varpi+\langle\ln\lambda_j\rangle+\sum_{s=1}^{j-1}\langle\ln(1-\lambda_s)\rangle\right]}{\sum_{j=1}^{M}\exp\left[\sum_{d=1}^{D}\langle\phi_{ijd}\rangle\xi+\sum_{d=1}^{D}\langle 1-\phi_{ijd}\rangle\varpi+\langle\ln\lambda_j\rangle+\sum_{s=1}^{j-1}\langle\ln(1-\lambda_s)\rangle\right]}\ , \\
&f_{ijd} = \frac{\exp\left[\langle Z_{ij}\rangle\xi+\langle\ln\epsilon_{jd_1}\rangle\right]}{\exp\left[\langle Z_{ij}\rangle\xi+\langle\ln\epsilon_{jd_1}\rangle\right]+\exp\left[\langle Z_{ij}\rangle\varpi+\langle\ln\epsilon_{jd_2}\rangle\right]}\ , \\
&\xi = \left[\widetilde{\mathcal{R}}_{jd} + (\bar{\alpha}_{jd}-1)\ln X_{id} + (\bar{\beta}_{jd}-1)\ln(1-X_{id})\right]\ , \qquad \theta_j = 1+\textstyle\sum_{i=1}^{N}\langle Z_{ij}\rangle\ , \\
&\varpi = \left[\widetilde{\mathcal{F}}_{jd}+(\bar{\sigma}_{jd}-1)\ln X_{id}+(\bar{\tau}_{jd}-1)\ln(1-X_{id})\right]\ , \qquad \vartheta_j = \langle\psi_j\rangle+\textstyle\sum_{i=1}^{N}\sum_{s=j+1}^{M}\langle Z_{is}\rangle\ , \\
&v_{jd}^{*} = v_{jd} - \textstyle\sum_{i=1}^{N}\langle Z_{ij}\rangle\langle\phi_{ijd}\rangle\ln X_{id}\ , \qquad a_j^{*} = a_j+1\ , \qquad b_j^{*} = b_j - \langle\ln(1-\lambda_j)\rangle\ , \\
&c_1^{*} = c_1 + \textstyle\sum_{i=1}^{N}\langle\phi_{ijd}\rangle\ , \qquad c_2^{*} = c_2 + \textstyle\sum_{i=1}^{N}\langle 1-\phi_{ijd}\rangle\ , \\
&u_{jd}^{*} = u_{jd}+\textstyle\sum_{i}\langle Z_{ij}\rangle\langle\phi_{ijd}\rangle\bar{\alpha}_{jd}[\Psi(\bar{\alpha}_{jd}+\bar{\beta}_{jd})-\Psi(\bar{\alpha}_{jd})+\bar{\beta}_{jd}\Psi'(\bar{\alpha}_{jd}+\bar{\beta}_{jd})(\langle\ln\beta_{jd}\rangle-\ln\bar{\beta}_{jd})]\ ,
\end{aligned}
$$

where $\Psi(\cdot)$ is the digamma function. Note that, $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{F}}$ are the lower bounds of $\mathcal{R} = \left\langle\ln\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right\rangle$ and $\mathcal{F} = \left\langle\ln\frac{\Gamma(\sigma+\tau)}{\Gamma(\sigma)\Gamma(\tau)}\right\rangle$, respectively. Since these expectations are intractable, we use the second-order Taylor series expansion to find their lower bounds as proposed in [18]. The solutions to the hyperparameters of $Q(\boldsymbol{\beta})$, $Q(\boldsymbol{\sigma})$ and $Q(\boldsymbol{\tau})$ can be computed similarly as for $\boldsymbol{u}^{*}$ and $\boldsymbol{v}^{*}$. The expected values in the above formulas are given by

$$\bar{\alpha}_{jd} = u_{jd}^{*}/v_{jd}^{*}\ , \qquad \bar{\beta}_{jd} = p_{jd}^{*}/q_{jd}^{*}\ , \qquad \bar{\sigma}_{jd} = g_{jd}^{*}/h_{jd}^{*}\ , \qquad \bar{\tau}_{jd} = s_{jd}^{*}/t_{jd}^{*} \tag{9}$$

$$\langle Z_{ij}\rangle\ = r_{ij}\ , \quad \langle\psi_j\rangle = a_j^{*}/b_j^{*}\ , \qquad \big\langle\phi_{ijd}\big\rangle\ = f_{ijd}\ , \qquad \langle\ln\beta\rangle\ = \Psi(p^{*}) - \ln q^{*} \tag{10}$$

$$\big\langle\ln\lambda_j\big\rangle = \Psi(\theta_j) - \Psi(\theta_j+\vartheta_j)\ , \qquad \big\langle\ln(1-\lambda_j)\big\rangle = \Psi(\vartheta_j) - \Psi(\theta_j+\vartheta_j) \tag{11}$$

$$\big\langle\ln\epsilon_{jd_1}\big\rangle = \Psi(c_1^{*}) - \Psi(c_1^{*}+c_2^{*})\ , \qquad \big\langle\ln\epsilon_{jd_2}\big\rangle = \Psi(c_2^{*}) - \Psi(c_1^{*}+c_2^{*}) \tag{12}$$

Since the solutions to each variational factor are coupled together through the expected values of other factors, the optimization of the model can be solved in a way analogous to the EM algorithm. First, we need to initialize the truncation level M and the values of all the hyperparameters. The variational E-step: estimate the expected values in (9)∼(12), use the current distributions over the model parameters. The variational M-step: update the variational solutions for each factor by (5) ∼ (8) using the current values of the moments. These two steps repeat until convergence criteria is reached. The optimal number of components M can be detected by eliminating the components with small mixing coefficients close to 0.

3 Experimental Results

In this section, we validate the proposed algorithm through two challenging real world applications, namely the web page clustering and the tissue sample categorization. The main goal of our experiments is to investigate the advantages of the proposed infinite GD mixture model with localized feature selection (*InLFsGD*) approach by comparing it to: the infinite GD mixture model with global feature selection (*InGFsGD*), the infinite GD mixture model without feature selection

(*InGD*), the finite GD mixture model with localized feature selection (*LFsGD*) proposed in [10] and the infinite Gaussian mixture model with localized feature selection (*InLFsGM*). To make a fair comparison, all of these methods are learned in a variational way. It is also noteworthy that all results are averaged over 20 runs of the algorithm. According to the experimental results that we have obtained, we initialize the proposed model as the following: the initial truncation level M is set to 15, the initial values of hyperparameters u, p, g and s of the Gamma priors are set to 1, v, q, h, t are set to 0.01, the hyperparameters a and b are set to 1, while c_1 and c_2 are both set to 0.1.

3.1 Web Page Clustering

In this experiment, the application of web page clustering is highlighted. In our case, a subset of the WebKB data set[1] is adopted, which is known as the WebKB4 data set. The WebKB4 consists of 4,199 web pages from the four most populous categories: student (1641 pages), faculty (1124 pages), course (930 pages), and project (504 pages). The methodology of our text clustering approach is decried as the following: First, the Rainbow package[2] was used as a preprocessing step to select the top 500 words by removing the rare (occurred less than 30 times) and stop words (such as “the”, “and”, “or”, etc.). Next, each web page was represented by a vector of counts (i.e. a histogram that containing the frequency of occurrence of each word in its vocabulary). Then, the latent Dirichlet allocation (LDA) model [4] was applied to reduce the dimensionality of those vectors. Accordingly, each document was represented by a vector of proportions. After applying the geometric transformation presented in Section 2, these vectors were then modeled by our infinite mixture using the algorithm in the previous section. Finally, the classification is performed by applying Bayes’ decision rule. Table 1 shows the confusion matrix of the WebKB4 data set obtained by applying the proposed *InLFsGD*. The average performances of web page clustering using different methods are shown in Table 2. As we can observed from this table, it is clear that the *InLFsGD* achieves the best performance among the five methods in terms of the highest classification accuracy rate (81.51%) and the most accurate estimated number of components (4.04).

Table 1. Average rounded confusion matrix calculated by *InLFsGD*

	Student	Faculty	Course	Project	Acc (%)
Student	**1381**	110	78	72	84.2
Faculty	135	**893**	51	45	79.4
Course	63	30	**796**	41	85.6
Project	41	49	27	**387**	76.8
			Overall Rate		81.5

[1] The data set is available at: http://www.cs.cmu.edu/~textlearning/

[2] http://www.cs.cmu.edu/~mccallum/bow/

Table 2. The average accuracy rate (Acc) (%) and the number of categories ($\widehat{M}$) obtained using different methods. The numbers in parenthesis are the standard deviation of the corresponding quantities.

Method	*InLFsGD*	*InGFsGD*	*InGD*	*LFsGD*	*InLFsGM*
$\widehat{M}$	4.04 (0.16)	4.21 (0.69)	4.97 (0.63)	5.23 (1.12)	4.78 (0.82)
Acc (%)	81.51 (1.56)	79.36 (1.32)	76.23 (1.75)	73.49 (2.01)	77.81 (1.55)

3.2 Tissue Sample Categorization

Clustering gene expression microarray data is a very challenging task and has received significant attention recently. One of the most effective technique for handling this clustering task is to adopt finite mixture models [19,14]. However, most of these approaches require some model selection criteria (ex. BIC, AIC, etc) to determine the number of components. In this experiment, we apply the proposed nonparametric variational approach for categorizing tissue samples which contain gene expressions. Unlike finite mixture model approaches mentioned above, our approach is able to bypass the problem of model selection and allows simultaneous separation of data in to similar clusters and selection of relevant features. In this case, we use the well-known Lymphoma data set which is introduced in [1] to measure the gene expression levels using Lymphochip microarrays. This data set contains 80 tissue samples and 4062 genes. Within those samples, there are 29 cases of of B-cell chronic lymphocytic leukaemia (B-CLL), 42 cases of diffuse large B-cell lymphoma (DLBCL) and 9 cases of follicular lymphoma (FL).

Table 3. The average accuracy rate (Acc) (%) and the number of categories ($\widehat{M}$) obtained using different methods

Method	*InLFsGD*	*InGFsGD*	*InGD*	*LFsGD*	*InLFsGM*
$\widehat{M}$	3.22 (0.23)	3.51 (0.46)	4.03 (0.95)	4.34 (1.23)	3.55 (0.71)
Acc (%)	86.15 (1.18)	85.02 (1.37)	82.39 (1.68)	79.08 (1.92)	84.36 (1.29)

In the preprocessing step, we normalized the data set into a range of $[0, 1]$ such that one feature (the gene repression levels in one particular array) would not dominant the others in our algorithm. Then, these vectors were learned using the proposed *InLFsGD*. Finally, the classification is performed by by assigning the tissue samples to the group which has the highest posterior probability according to Bayes' rule. Table 3 illustrates the clustering results acquired by employing different methods. According to the results in this table, it is clear again that infinite localized feature selection outperforms other methods.

4 Conclusion

In this paper, we have proposed a novel approach for simultaneous clustering and localized feature selection based on the variational learning of infinite GD mixture models (or Dirichlet processes of GD distributions) with localized feature selection. By using this model, the difficulty of determining the number of clusters is avoided and the problems of overfitting and underfitting are prevented. The effectiveness of the proposed approach has been evaluated on two real applications involving web pages clustering and tissue samples categorization.

References

1. Alizadeh, A.A., Eisen, M.B., Davis, R.E., et al.: Distinct Types of Diffuse Large B-cell Lymphoma Identified by Gene Expression Profiling. Nature 403, 503–511 (2000)
2. Attias, H.: A Variational Bayes Framework for Graphical Models. In: Proc. of Neural Information Processing Systems (NIPS), pp. 209–215 (1999)
3. Bishop, C.M.: Variational Learning in Graphical Models and Neural Networks. In: Proc. of ICANN, pp. 13–22. Springer (1998)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
5. Blei, D.M., Jordan, M.I.: Variational Inference for Dirichlet Process Mixtures. Bayesian Analysis 1, 121–144 (2005)
6. Bouguila, N., Ziou, D.: A Hybrid SEM Algorithm for High-Dimensional Unsupervised Learning Using a Finite Generalized Dirichlet Mixture. IEEE Transactions on Image Processing 15(9), 2657–2668 (2006)
7. Bouguila, N., Ziou, D.: High-Dimensional Unsupervised Selection and Estimation of a Finite Generalized Dirichlet Mixture Model Based on Minimum Message Length. IEEE Transactions on PAMI 29(10), 1716–1731 (2007)
8. Boutemedjet, S., Bouguila, N., Ziou, D.: A Hybrid Feature Extraction Selection Approach for High-Dimensional Non-Gaussian Data Clustering. IEEE Transactions on PAMI 31(8), 1429–1443 (2009)
9. Constantinopoulos, C., Titsias, M., Likas, A.: Bayesian Feature and Model Selection for Gaussian Mixture Models. IEEE Trans. on PAMI 28(6), 1013–1018 (2006)
10. Fan, W., Bouguila, N., Ziou, D.: Unsupervised Anomaly Intrusion Detection via Localized BayesianFeature Selection. In: Proc. of ICDM, pp. 1032–1037 (2011)
11. Fan, W., Bouguila, N., Ziou, D.: Variational Learning for Finite Dirichlet Mixture Models and Applications. IEEE Trans. Neural Netw. Learning Syst. 23(5), 762–774 (2012)
12. Ferguson, T.S.: Bayesian Density Estimation by Mixtures of Normal Distributions. Recent Advances in Statistics 24, 287–302 (1983)
13. Figueiredo, M., Jain, A.: Unsupervised Learning of Finite Mixture Models. IEEE Transactions on PAMI 24(3), 381–396 (2002)
14. Ji, Y., Wu, C., Liu, P., Wang, J., Coombes, K.R.: Applications of Beta-mixture Models in Bioinformatics. Bioinformatics 21(9), 2118–2122 (2005)
15. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An Introduction to Variational Methods for Graphical Models. Machine Learning 37(2), 183–233 (1999)
16. Law, M.H.C., Figueiredo, M.A.T., Jain, A.K.: Simultaneous Feature Selection and Clustering Using Mixture Models. IEEE Trans. on PAMI 26(9), 1154–1166 (2004)
17. Li, Y., Dong, M., Hua, J.: Simultaneous Localized Feature Selection and Model Detection for Gaussian Mixtures. IEEE Transactions on PAMI 31, 953–960 (2009)
18. Ma, Z., Leijon, A.: Bayesian Estimation of Beta Mixture Models with Variational Inference. IEEE Transactions on PAMI 33(11), 2160–2173 (2011)

19. McLachlan, G.J., Khan, N.: On a Resampling Approach for Tests on the Number of Clusters with Mixture Model-based Clustering of Tissue Samples. J. Multivar. Anal. 90(1), 90–105 (2004)
20. Neal, R.M.: Markov Chain Sampling Methods for Dirichlet Process Mixture Models. Journal of Computational and Graphical Statistics 9(2), 249–265 (2000)
21. Sethuraman, J.: A Constructive Definition of Dirichlet Priors. Statistica Sinica 4, 639–650 (1994)
22. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical Dirichlet Processes. Journal of the American Statistical Association 101, 705–711 (2004)

Generalized Agglomerative Fuzzy Clustering

Kiatichai Treerattanapitak and Chuleerat Jaruskulchai*

Department of Computer Science, Kasetsart University
50 Phahon Yothin Rd, Chatuchak Bangkok Thailand
kiat@smartzap.com, fscichj@ku.ac.th

Abstract. Fuzzy Cluster is a powerful for cluster analysis. However, inappropriate parameters selection leads Fuzzy Clustering to produce unreliable results. In addition, Fuzzy Clustering is sensitive to initialization and could be struck in local minima. Although, clustering results are validated by Cluster Validity Index but these methods obtain the best clustering result by reproduce clustering with various parameters and it is computation expensive. In order to overcome these issues, Generalized Agglomerative Fuzzy Clustering is proposed in this paper. Our proposed method is capable to find the optimum number of clusters and fuzzifier during the clustering execution. Moreover, this method is applicable to Fuzzy Clustering and its variants. Comprehensive experiments show that our agglomerative method obtained the right number of clusters and fuzzifier.

Keywords: Cluster validation, Fuzzifier, Parameter, Number of clusters.

1 Introduction

Clustering is a technique to separate unlabeled data into finite and discrete set. Traditional clustering like K-Means [1] puts each data into exactly one cluster. For overlapped datasets where some data can be allocated to multiple clusters, K-Means may not be the right method to analyze the dataset. To achieve better clustering, Fuzzy C-Mean (FCM) [2, 3] was proposed by incorporating with uncertainty to allow data being assigned of all clusters at different degree of membership. FCM requires predefined number of clusters and fuzzifier (fuzzy exponent) which do not know a priori. Even though, FCM is a powerful clustering algorithm but estimating right parameters is a difficulty that prevents FCM producing good quality [4-16]. In order to obtain an optimum result, FCM is executed with variation of input parameters. Major cluster characteristics of FCM results are compared among each other in term of compactness and separation by Cluster Validity Index (CVI). Compactness indicates the variation of the data within a cluster, and separation indicates the isolation of the clusters from each other. In addition, CVI is a tool to find the number of clusters which is a subject of cluster validity problem. Nevertheless, using CVIs require FCM to execute with the number of clusters starts from 2 and sequentially increase in later execution. This method consumes lot of efforts and computational

* Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 34–41, 2012.

resources. Furthermore, some CVIs are sensitive to noise and increase monotonically when number of clusters approaches number of data [16].

To overcome these drawbacks, we proposed an agglomerative FCM that determine number of clustering on the fly. Our method is capable to apply to any FCM variants. This paper is organized as follow. In section 2, we review existing CVI technique and agglomerative fuzzy clustering. In section 3, we propose agglomerative method by determining centroids for merging. In section 4, we evaluate our approach on real dataset. In Section 5, the conclusion is drawn with our future works.

2 Related Works

Fuzzy clustering i.e. FCM partitions a set of dataset δ ($x_i \in \delta$, $i=1..N$) into k clusters. It involves uncertainty i.e. the data could not belong to one cluster nevertheless all data belong to all clusters with the different degree of membership (μ_{ij}). The data is assigned to clusters by comparing its distance or dissimilarity (d_{ij}^2) to the cluster centroids (v_j). The optimum membership and centroid are obtained as (1) and (2).

$$\mu_{ij} = 1/\sum_{u=1}^{k}\left(\frac{{d_{ij}}^2}{{d_{iu}}^2}\right)^{\frac{1}{m-1}} \tag{1}$$

$$v_j = \sum_{i=1}^{N}\left({\mu_{ij}}^m x_i\right)/\sum_{i=1}^{N}{\mu_{ij}}^m \tag{2}$$

In some circumstances, FCM assign degree of membership too high level of fuzzy thus Exponential Fuzzy Clustering (XFCM) is FCM enhancement to improve membership assignment by reformulate objective function in exponential [4,17]. The optimum membership and centroid of XFCM are obtained as (3) and (4).

$$\mu_{ij} = e^{-m{d_{ij}}^2}/\sum_{u=1}^{k} e^{-m{d_{iu}}^2} \tag{3}$$

$$v_j = \sum_{i=1}^{N}\left(\mu_{ij} x_i\right)/\sum_{i=1}^{N}\mu_{ij} \tag{4}$$

Basically, FCM requires number of clusters (k) and fuzzifier (m) as input parameters to execute. Inappropriate parameters leads algorithm to produce poor clustering quality. In general, m is in range between [1.5,2.5][8] or setting to 2 in most cases [16]. The upper limit of m can be estimated using Eigen value of the matrix [5] and actual value can be estimated by a relation of number of data and dimensions [7]. However, FCM and XFCM produces crisp result (result similar to K-Means) if fuzzifier approaches 1. In reverse, if fuzzifier is too high, clustering produce average result i.e. the degrees of membership become $1/k$ [17]. To obtain optimum number of cluster, it is the subject of cluster validity problem. Most CVIs have their own benefits and weakness but fundamental approach is the same which based on separation and compactness. The most popularity CVI was proposed by Xie and Beni [9] which used the FCM's objective function as compactness. Kwon [10] extends the index of Xie and Beni by adding penalty term to eliminate monotonically decrease when number of clusters approaches number of data. Well-known of CVIs are summarized in Table 1.

Table 1. Well-known Cluster Validity Index

Xie and Beni [9]	$V_{XB} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{N}{\mu_{ij}}^{m}\lVert x_i - v_j\rVert^2}{N\left(\min_{i\neq}\lVert v_i - v_j\rVert^2\right)}$
Kwon [10]	$V_K = \frac{\sum_{i=1}^{k}\sum_{j=1}^{N}{\mu_{ij}}^{2}\lVert x_i - v_j\rVert^2 + \frac{1}{k}\sum_{i=1}^{k}\lVert v_i - v_j\rVert^2}{\min_{i\neq j}\lVert v_i - v_j\rVert^2}$
Fukuyama and Sugeno[11]	$V_{FS} = \sum_{i=1}^{k}\sum_{j=1}^{N}{\mu_{ij}}^{m}\lVert x_i - v_j\rVert^2 - N\left(\min_{i\neq j}\lVert v_i - v_j\rVert^2\right)$
Gath and Geve [12]	$V_{FHV} = \sum_{i=1}^{k}\left[\frac{\sum_{j=1}^{N}{\mu_{ij}}^{m}(x_i - v_j)(x_i - v_j)^T}{\sum_{j=1}^{N}{\mu_{ij}}^{m}}\right]^{1/2}$
Pakhira and Bandyopadhyay [13,14]	$V_{PMBF} = \left(\frac{\left(\sum_{j=1}^{N}\lVert x_i - \bar{x}\rVert\right)\left(\max_{i,j=1..k}\lVert v_i - v_j\rVert\right)}{k\sum_{i=1}^{k}\sum_{j=1}^{N}{\mu_{ij}}^{m}\lVert x_i - v_j\rVert}\right)^2$
Wu and Yang [15]	$V_{PCAES} = \sum_{i=1}^{k}\sum_{j=1}^{N}\frac{{\mu_{ij}}^{2}}{\min_{1\leq i\leq k}\sum_{j=1}^{N}{\mu_{ij}}^{2}} - \sum_{i=1}^{k}\exp\left(\frac{-\min_{i\neq .}\lVert v_i - v_j\rVert^2}{\sum_{i=1}^{k}\lVert v_i - \overline{x}\rVert^2/k}\right)$
Zhang [16]	$Var = \left(\frac{c+1}{c-1}\right)^{1/2}\sum_{i=1}^{k}\sum_{j=1}^{N}\frac{\mu_{ij}\left[1 - \exp\left(-\left(\sum_{i=1}^{N}\frac{\lVert x_i - \overline{x}\rVert^2}{n}\right)^{-1}\right)\right]}{N_k} \quad ,\overline{x} = \sum_{i=1}^{N}\frac{x_i}{N}$ $Sep = 1 - \max_{i\neq k}\left(\max_{x_j\in\delta}\min\ \left(\mu_{ij} - \mu_{kj}\right)\right)$ $V_w = \frac{Var/\max_{c=1..k}Var}{Sep/\max_{c=1..k}Sep}$

As aforementioned, obtaining number of clusters using CVI is computational expensive because of the long trail of clustering execution. Moreover, initialization can influence FCM and XFCM to local minima. In order to overcome these issues, agglomerative competitive fuzzy clustering (AFC) was proposed [18] and later improved using Entropy regularization [19]. These approaches start by over specify of number of clusters and repeatedly merge clusters by competitive compare cardinality of each cluster. Adjacent clusters with lower cardinality will be merged to clusters with higher cardinality where cardinality is the total degrees of membership of cluster. These algorithms do not sensitive to the selection of initialized seeds therefore these methods are applicable to specific algorithms [19]. In order to generalize these methods to other fuzzy variants, we propose centroid similarity as a merging condition for AFC.

3 Generalized Agglomerative Fuzzy Clustering

In general, Fuzzy clustering and its variants are similar in term of implementation. These methods begin with the setting of fuzzifier and number of clusters as input parameters. Fuzzy clustering is iteratively executed and outputs degree of membership and centroids. The main advantages of AFC method are that the number of clusters is obtained during the execution and clustering is not influenced by initialize and local minima. This method merges clusters from the over specify of number of cluster at start. The clusters with total degrees of membership lower than a specify threshold will be merged. Nevertheless, small clusters could be merged to the larger clusters [18]. Thus, in this paper we proposed the centroid similarity ($\delta(v_j,v_k)$) by comparing attributes (q where $q=1...Q$ and Q is the number of dimensions) against threshold ($\theta^2 Q$ where θ is [0,1]) as (5). In case of multiple centroids can be merged ($\delta(v_j,v_k)< \theta^2 Q$), the new centroid is simply computed by an average of each attribute. From our experiments, θ sets to 0.1-0.3 is generally used [19].

$$\delta(v_j, v_k) = \sum_{q=1}^{Q} \left(\frac{v_{jq}-v_{kq}}{\max(v_{jq}, v_{kq})}\right)^2, Threshold = \theta^2 Q \tag{5}$$

Basically, every fuzzy clustering algorithm has a target result indirectly indicating by fuzzifier i.e. Fuzzifier does not only represent the level of fuzziness, it represents how the target clustering will be. Fuzzifier is algorithm dependent. High and low value of fuzzifier cause the distribution of degree of membership (Var) in different behaviors. Unfortunately, existing CVIs do not take fuzzifier validation into account. From Fig.1(a), value of V_{XB} should minimize in order to represent a good clustering result while Normalized Variance ($NVar$) should maximize to avoid degree of membership to be average (the degrees of membership become $1/k$). Hence, the new index ($V_{XB}{}^{*}$) that capture fuzzifier should be tradeoff between these terms and new CVI that capture fuzzifier is defined as (6)

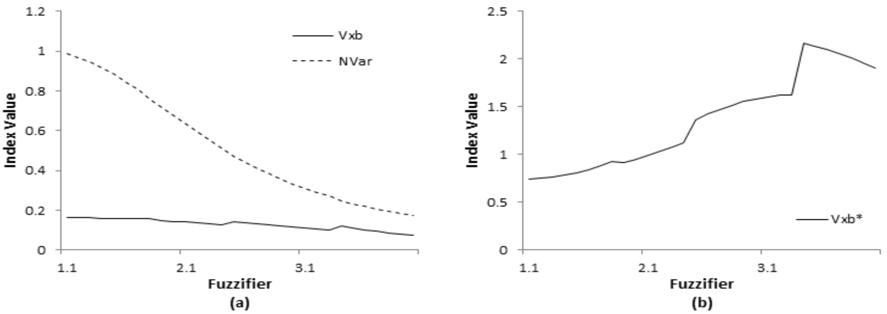


Fig. 1. FCM results from IRIS dataset with fuzzifier from 1.1-4.0. (a) Changes of V_{XB} and $NVar$ against fuzzifier and (b) Tradeoff result of V_{XB} and $NVar$ against fuzzifier ($V_{XB}{}^{*}$).

$$V_{XB}{}^{*} = NVar * V_{XB}, \quad NVar = \frac{Var}{MaxVar}, \quad Var = \frac{\sum_{j=1}^{k}\sum_{i=1}^{N}(\mu_{ij}-\overline{\mu_{\iota J}})}{Nk-1}$$
$$\overline{\mu_{\iota J}} = \sum_{j=1}^{k}\sum_{i=1}^{N}\frac{\mu_{ij}}{Nk}, \quad MaxVar = \frac{N\left(1-\frac{1}{k}\right)^2+\frac{N(k-1)}{k^2}}{Nk-1} = \frac{Nk(k-1)}{k^2(Nk-1)} \tag{6}$$

The new AFC procedure consists of three steps as illustrated by Fig. 2. In step 1, the algorithm required k to be larger than actual number of clusters, initial fuzzifier to any number greater than 1 (e.g. m=1.1) and termination condition to a small number (e.g. ε=1). In step 2, normal fuzzy clustering is executed and clusters are merged according to centroid similarity at the end of iteration. This step ensures that the optimum number of clusters is obtained. The first three processes are to compute similarity, degree of membership and centroid. These processes are general steps of FCM and its variants, it costs $O(Nk)$. For the centroid similarity computation and merging centroids at the end of step 2, these process would cost *O(k log k)* and it is always lower than *O(Nk)*. In the last process of step 2 which repeat all process again. This repetition is also similar to FCM. Assume that it requires *l* iterations, thus time complexity for step 2 would be *O(Nkl)* which is the same as FCM. For FCM to obtain the optimum *k*, it requires *O(Nklq)* where *q* is the iteration process on top of normal clustering procedure. Hence, AFC uses lower time complexity. In step 3, *m* is increased and process fuzzy clustering until value of $V_{XB}{}^{*}$ is decreased (see Fig.1(b)). This step ensures that the optimum fuzzifier is obtained. It processes on top of step 2. Assume it requires *p* iterations, thus it is *O(Nklp)* for the algorithm to get both optimum *k* and *m*. However the AFC is very flexible, it can be processed by discarding step 3 or apply only step 2 or step 3 to other FCM and its variants procedure.

```
Step 1: Input $k$, $\theta$, m=1.1 and $\varepsilon$ (Terminate coefficient).
Step 2: Execute Agglomerative Fuzzy Clustering.
- Compute similarity between data and centroids.
- Compute degree of membership of each data against centroids.
- Compute centroids based on degree of membership.
- Compute centroid similarity as (5) and compare with threshold.
- Merge centroid and update the number of clusters.
- Repeat Step 2 until Terminate condition is met.
Step 3: Obtain optimum fuzzifier
- Set m=m+0.1, k=optimum number of clusters from Step 2.
- Perform normal Fuzzy Clustering
- Calculate Vxb* as (6)
- Repeat Step 3 and return output if Vxb* begins to drop.
```

Fig. 2. Procedure of Fuzzy Clustering using CVI with fuzzifier effect

4 Experiments

We implement AFC on FCM (AFCM) and XFCM (AXFCM) and validate their performance. First, we perform clustering using AFC (*k*=10 is used for initial number of clusters) to compare optimum centroids produce from each algorithm by initialization with the true centroids of dataset. Second, we compare the number of clusters with popular CVIs in Table 1. Four real datasets from UCI and 2 synthetics dataset are used in these experiments as summarized in Table 2.

Table 2. Dataset information for experiments

Dataset	Information
IRIS	150 instances, 3 classes, 4 attributes
Wine	178 instances, 3 classes, 13 attributes
Diagnostic Breast Cancer (WDBC)	569 instances, 2 classes, 30 attributes
Prognostic Breast Cancer (WPBC)	198 instances, 2 classes, 32 attributes
Synthetic data E4	184 instances, 4 classes, 2 attributes
Synthetic data E2	202 instances, 2 classes, 2 attributes

4.1 Experiment 1

In this experiment, we perform AFC and normal fuzzy clustering (using m=2) on IRIS and 2 synthetics datasets E2 and E4. The dataset (E4) consists of 4 clusters with 46 instances in each clusters. This dataset is generated by uniform distribution within the boundary of non-overlapping of 4 ellipse shapes which defined as (7). The dataset (E2) consists of 2 clusters with 101 instances in each cluster. The data points are uniformly selected within the boundary of two overlapped clusters which defined as (8).

$$\begin{aligned}&(x-11)^2/12.25+(y-11)^2/9=1, \quad (x-4)^2/9+(y-4)^2/9=1\\&(x-11)^2/4+(y-3.3)^2/9=1, \quad (x-3.5)^2/9+(y-11)^2/9=1\end{aligned} \tag{7}$$

$$(x-10)^2/49+(y-10)^2/9=1, \quad (x-4)^2/9+(y-9)^2/49=1 \tag{8}$$

We execute clustering 10 times per dataset and measure the centroids errors by comparing the produced centroids with the true centroids using MAE as defined in (9). The true centroid from IRIS obtained from Kothari et, al. [6].

$$MAE = \sum_{i=1}^{Q} \left| v_{jq} - v_{kq} \right| / Q \tag{9}$$

Table 3. MAE and Variance produces from FCM, XFCM, AFCM and AXFCM

Dataset	FCM	XFCM	AFCM	AXFCM
IRIS	0.21	0.13	0.10	0.07
E2	11.23	12.42	2.98	12.27
E4	19.91	18.16	14.81	11.33

From Table 3, the results show that Generalized Agglomerative method that applies to both FCM and XFCM yield the centroid errors less than original FCM and XFCM. One reason is that the number of initialization uses in Agglomerative method is larger than actual number of clusters. Consequently, the opportunity to get data from valid clusters is increased. In addition, some centroid errors produced on IRIS is jump to 0.6 from 0.03 for FCM and XFCM.

4.2 Experiment 2

We perform AFCM and AXFCM to obtain the number of clusters on real 4 datasets comparing to the results of FCM that validate by CVIs in Table 1.

Table 4. Optimum Number of clusters from AFCM, AXFCM comparing to other CVIs [16]

Dataset	k^*	AFCM	AXFCM	V_{XB}	V_K	V_{FS}	V_{FHV}	V_{PMBF}	V_{PCAES}	V_W
IRIS	3	3	3	2	2	5	3	3	2	3
Wine	3	3	3	3	3	13	3	3	3	3
WDBC	2	2	2	2	2	12	2	2	2	2
WPBC	2	2	2	2	2	4	2	2	2	2

From Table 4, both AFCM and AXFCM return the number of clusters as same as actual number of clusters (k^*) and better than other well-known CVIs sometimes. In addition, time consumed by AFCM and AXFCM (see Table 5) is a bit lower than FCM and XFCM except E2. The lower usage time is from the number of similarity computation decreases when clusters are merged by Agglomerative method. On the other hand, FCM and XFCM need longer time than Agglomerative method before converge. It is noted that FCM and XFCM required multiple execution of clustering process in order to obtain the right number of clusters while AFCM and AXFCM return the right number of clusters by single execution.

Table 5. Execution time in seconds by FCM, XFCM, AFCM and AXFCM

Dataset	FCM	XFCM	AFCM	AXFCM
E2	10	9	13	12
E4	20	18	16	11
IRIS	10	13	6	8
Wine	100	65	50	33
WDBC	106	70	61	66
WPBC	46	42	32	39

Table 6. Fuzzifier value obtained from AFCM

Dataset	E2	E4	IRIS	Wine	WDBC	WPBC
AFCM	4.0	2.9	1.8	1.3	1.1	3.2
Estimation [7]	8.5	8.7	3.2	1.3	1.1	1.1

We compare fuzzifier value obtained from AFCM with the esimation equation proposed by Schwammle [7]. For Wine and WDBC, both methods yield the same result. However, the fuzzifier obtained from Estimation for E2 and E4 are too high. At these fuzzifier values, most membership produced from algorithm approaches *1/k* which does not seem right comparing to the structure of dataset in Fig.3 therefore AFCM is more reliable.

5 Conclusion

Select the right value of fuzzifier and number of clusters parameters is crucial for fuzzy clustering. The number of clusters are usually obtained by validating the clustering result with Cluster Validity Index. Nevertheless, this method does not take fuzzifier into account. Even though, there exist a general recommendation for fuzzifier but the right value of fuzzifier for particular dataset is difficult to estimate. We propose the Generalized Agglomerative Fuzzy Clustering that is applicable to Fuzzy Clustering variants. We demonstrate Generalized Agglomerative Fuzzy Clustering by applying to FCM and XFCM and validate their performance with various experiments. The results show that Agglomerative Fuzzy Clustering obtained number of clusters and select the right fuzzifier during the execution. However, this method could be improved by studying the threshold in the relation with initial seeds and dataset by statistical reasoning and it is the area of future improvement.

References

1. MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceeding of fifth Berkeley Symposium on Math. Stats. and Probability, vol. 1, pp. 281–297 (1967)
2. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. J. Cybern. 3, 32–57 (1973)
3. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algoritms. Plenum Press, New York (1981)
4. Treerattanapitak, K., Jaruskulchai, C.: Membership Enhancement with Exponential Fuzzy Clustering for Collaborative Filtering. In: Proceeding of the 17th International Conference on Neural Info. Processing, pp. 559–566. Springer, Heidelberg (2010)
5. Yu, J., Cheng, Q., Huang, H.: Analysis of the Weighting Exponent in the FCM. IEEE Trans. Syst. Man, Cybern. B. 34, 634–639 (2004)
6. Kothari, R., Pitts, D.: On Finding the Number of Clusters. Pattern Recogn. Lett. 20, 405–416 (1999)
7. Schwammle, V., Jensen, O.N.: A Simple and Fast Method to Determine the Parameters for Fuzzy c–means Cluster Analysis. J. Bioinform. 26, 1–8 (2010)
8. Wu, K.: Analysis of Parameter Selections for Fuzzy c-means. Pattern Recogn. 45, 407–415 (2012)
9. Xie, X.L., Beni, G.: A Validity Measure for Fuzzy Clustering. IEEE Trans. Pattern Anal. Mach. Intell. 13, 841–847 (1991)
10. Kwon, S.H.: Cluster Validity Index for Fuzzy Clustering. Electron. Lett. 34, 176–217 (1998)
11. Fukuyama, Y., Sugeno, M.: A New Method of Choosing the Number of Clusters for the Fuzzy c-means Method. In: Proceedings of Fifth Fuzzy Systems Symposium, pp. 247–250 (1989)
12. Gath, I., Geva, A.B.: Unsupervised Optimal Fuzzy Clustering. IEEE Trans. Pattern Anal. Mach. Intell. 11, 773–781 (1989)
13. Pakhira, M.K., Bandyopadhyay, S., Maulik, U.: Validity Index for Crisp and Fuzzy Clusters. Pattern Recogn. 37, 487–501 (2004)
14. Pakhira, M.K., Bandyopadhyay, S., Maulik, U.: A Study of Some Fuzzy Cluster Validity Indices, Genetic Clustering and Application to Pixel Classification. Fuzzy Sets Syst. 155, 191–214 (2005)
15. Wu, K.L., Yang, M.S.: A Cluster Validity Index for Fuzzy Clustering. Pattern Recogn. Lett. 26, 1275–1291 (2005)
16. Zhang, Y., Wang, W., Zhang, X.: A Cluster Validity Index for Fuzzy Clustering. Inform. Sci. 178, 1205–1218 (2008)
17. Treerattanapitak, K., Jaruskulchai, C.: Exponential Fuzzy C-Means for Collaborative Filtering. J. Comput. Sci. Tech. 27, 567–576 (2012)
18. Frigui, H., Krishnapuram, R.: Clustering by Competitive Agglomeration. Pattern Recogn. 30, 1109–1119 (1997)
19. Li, M.J., Ng, M.K., Cheung, Y., Huang, J.Z.: Agglomerative Fuzzy K-Means Clustering Algorithm with Selection of Number of Clusters. IEEE Trans. Know. Data Eng. 20, 1519–1534 (2008)

A Novel Self-Adaptive Clustering Algorithm for Dynamic Data

Ming Liu*, Lei Lin, Lili Shan, and Chengjie Sun

MOE-MS Key Laboratory of Natural Language Processing and Speech,
School of Computer Science and Technology,
Harbin Institute of Technology, Harbin, China
{mliu,linl,shanlili,cjsun}@insun.hit.edu.cn

Abstract. Along with the fast advance of internet technique, internet users have to deal with novel data every day. For most of them, one of the most useful knowledge exploited from web is about the transfer of the information expressed by dynamically updated data. Unfortunately, traditional algorithms often cluster novel data without considering the existent clustering model. They have to cluster input data over again, once input data are updated. Hence, they are time-consuming and inefficient. For efficiently clustering dynamic data, a novel **S**elf-**A**daptive **C**lustering algorithm (abbreviated as SAC) is proposed in this paper. SAC comes from **S**elf **O**rganizing **M**apping algorithm (abbreviated as SOM), whereas, it doesn't need to make any assumption about neuron topology beforehand. Besides, when input data are updated, its topology remodels meanwhile. Experiment results demonstrate that SAC can automatically tune its topology along with the update of input data.

Keywords: Self-adaptive algorithm, Competitive learning, Minimum spanning tree, Self-organizing-mapping.

1 Introduction

Due to the fast advance of internet technique, the data from web are unstable and dynamically updated at times (this kind of data is denoted as dynamic data in this paper). This phenomenon forces internet users to face to novel data anywhere and anytime. In general, via clustering dynamic data, it is easy to acquire the knowledge about, what information appears, what information disappears, and what information maintains. This kind of knowledge is essential to the men who need to make the decisions via observing dynamic data.

As indicated by [1], there have been proposed many methods to cluster dynamic data as followings.

Dhillon *et al* in [2] just propose a dynamic clustering algorithm to help analyze the transfer of information. Unfortunately, this algorithm is time-consuming and impractical, since it needs to run several times. Ghaseminezhad and Karami in [3] improve this algorithm by employing SOM structure, which forms an initial neuron topology at first and then dynamically tunes its topology once input data are updated.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 42–49, 2012.

In order to enable neuron topology easily to be altered, some self-adaptive algorithms have been proposed. The prominent merit of them is that they don't need to set any assumption about neuron topology in advance. For example, Melody in [4] initializes a neuron topology of small scale at first and then gradually expands it following the update of input data. Tseng *et al* in [5] improve this algorithm by tuning neuron topology in virtue of dynamically creating and deleting arcs between neurons.

Unfortunately, aforementioned self-adaptive algorithms have two defects. One is that, when neuron topology isn't suitable for current input data, these algorithms will insert or split neurons, whereas, these newly created neurons may locate out of the area where input data distribute. The other is that, these algorithms fail to preserve topology order. Therefore, they can't perform competitive learning as transitional SOM based algorithms, which will generate some dead neurons and they will never be tuned. The detailed discussions are indicated in [6, 7].

For effectively clustering dynamic data, a novel **S**elf-**A**daptive **C**lustering algorithm (abbreviated as SAC) is proposed in this paper. Its neuron topology can be dynamically tuned following the update of input data. Moreover, it employs local density to create neurons, and imports minimum spanning tree to perform competitive learning. Experiments demonstrate that, our algorithm earns better performance than most of traditional topology fixed and topology self-adaptive algorithms.

2 Self-Adaptive Clustering Algorithm (SAC)

In this paper, we design a novel self-adaptive clustering algorithm. It links neurons by arcs, and dynamically creates and deletes links to enable this neuron topology easily to be altered. Besides, it employs local density to create new neurons to avoid "some neurons out of the area where input data distribute", and imports minimum spanning tree to perform competitive learning to avoid "dead neuron".

2.1 Neuron Creation

As indicated by [8], the general way to expand neuron topology is to combine the neuron which has the largest accumulation error with its least similar neighbor to form a new one. Unfortunately, this plan brings an inconvenient consequence that it may create the neurons which locate out of the area where input data distribute.

For dealing with it, SAC imports local density, proposed by Duan *et al* in [9], to create new neuron.

Let D_i represent one datum among input data. The neuron which is created from D_i is marked as N_i. The process that creates N_i from D_i by local density is as follows:

Choose t samples from input data, where t is decided by user. Among them, the kth sample has kth similarity to D_i. Calculate local density of D_i and each sample among those t data by

$$LocalDensity(SD_{ik}) = \frac{\sum_{r=1}^{m} \frac{Density(SD_{ik})}{Density(Neighbor(SD_{ik}, r))}}{m} \tag{1}$$

where, SD_{ik} represents the datum which has kth similarity to D_i. $Neighbor(SD_{ik},r)$ represents the datum which has rth similarity to SD_{ik}. m represents the quantity of the neighbors which are adjacent to SD_{ik}, and it often equals to t. $Density(SD_{ik})$ represents the density of the district around SD_{ik}, and can be calculated by

$$Density(SD_{ik}) = \frac{\sum_{r=1}^{m} | SD_{ik} - Neighbor(SD_{ik},r) |^2}{m} \tag{2}$$

2.2 Training Process of SAC

Due to lacking of topology order, the adjacent neurons of the winner neuron (the neuron which has the maximal similarity to training sample) can't be found by traditional self-adaptive algorithms. Thus, they can't perform competitive learning as indicated by [10]. Since our algorithm is also a kind of self-adaptive algorithms, we employ minimum spanning tree to form topology order as performed by [11].

In virtue of minimum spanning tree, we can define neuron adjustment range in the following equation, which is never carried out by traditional self-adaptive algorithms.

$$\delta(m,t) = a(t) * \max_{N_b \in \varepsilon}(| N_{\mathrm{m}} - N_b |^2) \tag{3}$$

where, N_m represents the winner neuron. ε represents the set which includes the neurons that are directly connected to N_m in minimum spanning tree, such as N_b. $a(t)$ is learning rate which monotonously drops along with training process [12].

By means of neuron adjustment range, we can tune neurons by

$$N_b(t+1) = N_b(t) + a(t) * \exp(-\frac{R_{bm}}{2\delta^2(m,t)}) * [D_i - N_b(t)]; \quad N_b \in \delta(m,t) \tag{4}$$

where, N_m represents the winner neuron which has the maximal similarity to D_i. t represents the index of training steps. R_{bm} is relation value between N_m and N_b. It indicates the weight of the link between N_m and N_b, and can be acquired by

$$R_{bm} = \frac{Sim_{bm}}{\sqrt{\sum_{p=1}^{C}\sum_{q=1}^{C} Sim_{pq}{}^2}} \tag{5}$$

where, C represents the quantity of clusters, which equals to the number of neurons. Sim_{pq} represents the similarity between two neurons, such as N_p and N_q.

$$Sim_{pq} = \sum_{k=1}^{z} | W_{pk} - W_{qk} |^2 * \frac{LocalDensity(N_p)}{LocalDensity(N_q)} \tag{6}$$

where, z represents the dimension of neuron vector. W_{pk} represents the weight of kth entry in N_p. So is to W_{qk}.

So, why we choose the neurons, which are directly connected to the winner neuron, to form neuron's adjustment range? The reason is that, the neurons which are directly connected to the winner neuron are more similar to the winner neuron than to other neurons. Training process of SAC is listed as follows:

1. Let *NEURON* represent neuron set. Let *LINK* represent link set. Each link has two parameters. One of them is relation parameter to indicate its weight. The other is age parameter to denote its creating time. Initialize error coefficient of each neuron with 0. Let *Inset* represent data set. Let *t* represent the index of training steps.

2. Randomly choose a datum from *Inset*, and mark it as D_i. Calculate the similarity between D_i and each neuron in *NEURON* by Eq.6.

3. Choose the neuron which has the maximal similarity to D_i as the winter neuron, and mark it as *MBN*. Tune *MBN* and its adjacent neurons by Eq.4. Increase error coefficient of *MBN* by

$$err_m = err_m + | D_i - MBN |^2 \tag{7}$$

where, err_m represents error coefficient of *MBN*.

4. Choose the neuron which has the secondly maximal similarity to D_i, and mark it as *SBN*. If there is no link between *MBN* and *SBN*, go to 5. If not, go to 6.

5. Create a link between *MBN* and *SBN*, mark it as l_{ms}, and insert it in *LINK*.

6. Apply Eq.5 to calculate relation parameter-R_{ms} of l_{ms}. Assign age parameter-Age_{ms} of l_{ms} with 0.

7. Add 1 to age parameter of each link in *LINK*.

8. Check each link in *LINK*. If there is a link whose age parameter is beyond the average value of all the links, remove it.

9. Check each neuron in *NEURON*. If there is a neuron which isn't connected by any link, remove it.

10. Increase *t* to *t*+1.

11. Let *s* denote quantity of input data. If (*t* mod *s*) equals 0, go to 12, else, go to 2.

12. Choose the neuron which has the maximal error coefficient, and mark it as N_q. Choose the neuron which is adjacent to N_q and has the minimal similarity to N_q, and mark it as N_f.

13. Combine N_q and N_f to create a new neuron by

$$N_r = \frac{N_q + N_f}{2} \tag{8}$$

Mark this newly created neuron as N_r. Create the links between N_q and N_r, N_f and N_r, and insert them in *LINK*. Initialize age parameter of each created link with 0.

14. Reduce error coefficients of N_q and N_f by

$$err_q = err_q / 2 \tag{9}$$

$$err_f = err_f / 2 \tag{10}$$

Assign N_r with new error coefficient by

$$err_r = \frac{err_q + err_f}{2} \tag{11}$$

15. Check whether neuron topology has met convergence condition or not. If yes, stop. If not, go to 2.

2.3 The Process to Cluster Dynamic Data

For helping explain how to use SAC to cluster dynamic data, let's adopt some symbols. Let t_1 and t_2 represent two time phases. Let $InSett_1$ and $InSett_2$ represent the data sets respectively collected in t_1 and t_2. For clustering the dynamic data from t_1 to t_2, we firstly use SAC to form a neuron topology according to $InSett_1$. When $InSett_1$ is updated, for example, changing to $InSett_2$, we use SAC to alter the existent neuron topology according to $InSett_2$. The altering process is just the same to the training process of SAC. Once data set is updated again, it only needs to run this training process once more to alter the existent neuron topology according to the updated data set.

3 Experiments and Analyses

3.1 Experiments on Clustering Performance

As indicated by [13], UCI data set is one of the most prevalent testing corpora for clustering algorithms. Since it contains too many kinds of data sets, we only select some extensively applied data sets as the standard testing corpus to compare the performance of SAC with that of the other clustering algorithms in Table 1. They are GNG [14] PSOM [15], DASH [16], SOM [17], GSOM [18], and GHSOM [19].

In the following experiments, we employ *Purity* in [20] as the measurement, which can be calculated by

$$Purity = \sum_{r=1}^{z} \frac{n_r}{n} P(S_r) \tag{12}$$

where, z represents the quantity of clusters. n represents the quantity of input data. S_r represents rth cluster formed by clustering algorithm. n_r represents the quantity of the data included by S_r. $P(S_r)$ can be calculated by

$$P(S_r) = \frac{1}{n_r} \max_{q=1}^{z} (n_r^q) \tag{13}$$

In UCI, it already partitions testing data into some predefined clusters. Let C_q represent qth cluster among the predefined clusters. Let n^q represent the quantity of the data included by C_q. Let $n_r^q = n^q \bigcap n_r$, which represents the quantity of the data, belonging to C_q in testing corpus and belonging to S_r after clustering algorithm.

Table 1. Purities of different clustering algorithms on the selected data sets

#DATA SETS/METHODS	SOM	GSOM	GHSOM	GNG	PSOM	DASH	SAC
Thyroid Gland	78.37	79.64	81.25	77.56	76.22	78.38	**82.19**
Japanese Credit Approval	74.18	76.36	79.94	74.93	73.48	77.50	**80.86**
Wine Recognition	75.32	77.32	81.04	76.58	75.83	80.21	**81.56**
Breast Cancer	76.54	78.43	80.07	73.61	74.69	79.54	**82.04**
Iris	81.12	82.01	83.25	78.82	76.31	82.07	**84.59**
Sonar Target	75.40	77.51	79.37	76.80	75.65	78.93	**80.31**
Ionosphere	78.33	80.17	81.95	79.56	77.30	80.03	**82.73**
Heart Disease	72.09	73.24	75.81	72.51	69.33	74.96	**77.20**
Waveform	70.66	71.83	73.87	70.91	69.07	73.09	**75.11**
Pima Diabetes	74.59	78.11	80.34	76.82	74.33	79.22	**81.55**
Multiple Feature	71.60	75.88	77.93	72.11	70.39	76.78	**78.27**
Optical Digit	78.62	80.13	82.31	79.58	77.54	80.43	**82.68**
German Credit Approval	72.17	73.45	75.71	73.36	70.08	72.69	**77.08**
Car Evaluation	76.76	79.86	81.65	78.65	76.34	80.77	**82.51**

Obviously, SAC has the best performance than any other clustering algorithm. This is because, it doesn't need to make any assumption about neuron topology beforehand, and can dynamically form neuron topology to simulate the distribution of input data. Besides, to further boost its performance, it constructs minimum spanning tree to perform competitive learning. Through pervious operations, SAC consequently earns the best performance.

3.2 Experiments on Dynamic Data

The data from UCI data set don't change along with time passing. Therefore, we can't utilize them to test the performance of SAC for dynamic data. For this reason, we crawl ten thousands news web-pages from website over the entire year of 2010 as testing corpus, and separate it into four sets to represent the dynamic data collected in four time phases. Let $InSett_1$, $InSett_2$, $InSett_3$, $InSett_4$ represent the data sets respectively including the news from January to March, from April to June, from July to September, from October to December. Clustering results are shown in Table 2.

Table 2. Purities of different clustering algorithms based on the existent neuron topology

#DATA SETS/METHODS	SOM	GSOM	GHSOM	GNG	PSOM	DASH	SAC
January to March	75.37	74.64	76.25	74.56	72.22	75.38	**81.19**
April to June	-	-	-	68.33	66.48	69.71	**79.24**
July to September	-	-	-	63.23	61.67	64.31	**78.13**
October to December	-	-	-	58.76	56.39	59.25	**77.42**

As Table 2 shows, traditional topology fixed algorithms can't perform clustering on dynamic data, and traditional topology self-adaptive algorithms can cluster dynamic data based on the existent neuron topology. However, due to lacking of solutions to deal with "dead neuron" and "neurons locating out of area where input data distribute", when input data update, the performances of traditional topology self-adaptive algorithms drop sharply. Correspondingly, SAC keeps its high performance. This is because, SAC imports density to create new neurons, and imports minimum spanning tree to perform competitive learning. Besides, the method adopted by SAC via creating and deleting links between different neurons can make neuron topology easily to be altered along with the update of input data.

4 Conclusion

Along with the fast advance of internet technique, novel data appear every day. In order to cluster them, a novel self-adaptive clustering algorithm is proposed in this paper, which is abbreviated as SAC. This algorithm doesn’t need to make any assumption about neuron topology in advance, and can dynamically form it to simulate the distribution of input data. For avoiding neurons from locating out of the area where input data distribute, it adopts local density to create new neurons. Besides, minimum spanning tree is imported to perform competitive learning to further enhance its performance. Experiment results demonstrate that SAC works better than most of traditional clustering algorithms. It can cluster dynamic data very well.

Acknowledgements. The research in this paper is supported by “the Fundamental Research Funds for the Central Universities” (Grant No. HIT. NSRIF. 2013066) and Key Laboratory Opening Funding of China MOE-MS Key Laboratory of Natural Language Processing and Speech.

References

1. Martin, S., Detlef, N.: Towards the Automation of Intelligent Data Analysis. Appl. Soft Comput. 6, 348–356 (2006)
2. Dhillon, I.S., Guan, Y.Q., Kogan, J.: Iterative Clustering of High Dimensional Text Data Augmented by Local Search. In: Proceedings of the Second IEEE International Conference on Data Mining, pp. 131–138. IEEE Press, Japan (2002)
3. Ghaseminezhad, M.H., Karami, A.: A Novel Self-Organizing Map (SOM) Neural Network for Discrete Groups of Data Clustering. Appl. Soft Comput. 11, 3771–3778 (2011)
4. Melody, Y.K.: Extending the Kohonen Self-Organizing Map Networks for Clustering Analysis. Comput. Stat. Data Anal. 38, 161–180 (2001)
5. Tseng, C.L., Chen, Y.H., Xu, Y.Y., Pao, H.T., Fu, H.C.: A Self-Growing Probabilistic Decision-Based Neural Network with Automatic Data Clustering. Neurocomput. 61, 21–38 (2004)
6. Melody, Y.K.: Extending the Kohonen Self-Organizing Map Networks for Clustering Analysis. Comput. Stat. Data Anal. 38, 161–180 (2001)
7. Lee, S., Kim, G., Kim, S.: Self-Adaptive and Dynamic Clustering for Online Anomaly Detection. Expert Syst. Appl. 38, 14891–14898 (2011)
8. Hodge, V.J., Austin, J.: Hierarchical Growing Cell Structures: TreeGCS. IEEE Trans. Knowl. Data Engin. 13, 207–218 (2001)
9. Duan, L., Xu, L.D., Guo, F., Lee, J., Yan, B.P.: A Local-Density Based Spatial Clustering Algorithm with Noise. Inform. Syst. 32, 978–986 (2007)

10. Ezequiel, L.R.: Probabilistic Self-Organizing Maps for Qualitative Data. Neural Networks 23, 1208–1225 (2010)
11. Tokunaga, K., Furukawa, T.: Modular Network SOM. Neural Networks 22, 82–90 (2009)
12. Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Paatero, V., Saarela, A.: Self Organization of a Massive Document Collection. IEEE Trans. Neural Networks 11, 574–585 (2000)
13. Blake, C., Keogh, E., Merz, C.J.: UCI Repository of Machine Learning Databases. University of California, Irvine (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
14. Alahakoon, D., Halganmuge, S.K., Srinivasan, B.: Dynamic Self-Organizing Maps with Controlled Growth for Knowledge Discovery. IEEE Trans. Neural Networks 11, 601–614 (2000)
15. Rauber, A., Merkl, D., Dittenbach, M.: The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High-Dimensional Data. IEEE Trans. Neural Networks 13, 1331–1341 (2002)
16. Qin, A.K., Suganthan, P.N.: Robust Growing Neural Gas Algorithm with Application in Cluster Analysis. Neural Networks 17, 1135–1148 (2004)
17. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (1995); (2nd Extended Edition 1997)
18. Robert, L.K., Warwick, K.: The Plastic Self Organising Map. In: Proceedings of the 2002 International Joint Conference on Neural Networks, pp. 727–732. IEEE Press, Hawaii (2002)
19. Hung, C., Wermter, S.: A Dynamic Adaptive Self-Organising Hybrid Model for Text Clustering. In: Proceedings of the Third IEEE International Conference on Data Mining, pp. 75–82. IEEE Press, Melbourne (2003)
20. Gu, M., Zha, H., Ding, C., He, X.: Simon, H., Xia, J.: Spectral Relaxation Models and Structure Analysis for K-Way Graph Clustering and Bi-Clustering. Technical Report, CSE-01-007, Penn State University (2001)

Impulsive Synchronization of State Delayed Discrete Complex Networks with Switching Topology

Chaojie Li[1], David Yang Gao[1], and Chao Liu[2]

[1] School of Science, Information Technology and Engineering,
University of Ballarat, Mt Helen,VIC 3350, Australia
[2] College of Computer, Chongqing University, 400040, P.R. China
cjlee.cqu@163.com

Abstract. In this paper, global exponential synchronization of a class of discrete delayed complex networks with switching topology is investigated by using Lyapunov-Ruzimiki method. The impulsive scheme is designed to work at the time instant of switching occurrence. A time-varying delay dependent criterion for impulsive synchronization is given to ensure the delayed discrete complex networks switching topology tending to a synchronous state. Furthermore, a numerical simulation is given to illustrate the effectiveness of main results.

Keywords: Complex networks, impulsive synchronization.

1 Introduction

It has long been understood that many physical, social, biological, and technological networks are modeled by a graph with non-trivial topological features. In this model, every node is an individual element of the whole system with certain pattern of connections, in which connections between each pair of nodes are neither entirely regular nor entirely random[1],[2],[3].Secure communication[4],[5],parallel image processing[6] and chemical reaction implemented by coupled chaotic systems have been an active research field during the last two decades. As a consequence, theory and methods for synchronization of different families of complex networks have been extensively studied by many researchers(such as,[7]–[10]) and references therein). The improvement on different regimes of synchronization of discrete complex networks are abstracted from papers authored by [9]. Some general cases of synchronization of complex networks with switching topology can be found in the literatures of [14]. Adaptive synchronization, impulsive synchronization scheme and pining control synchronization have been considered by authors in[14]-[16]. Impulsive control has been successfully used to stabilize and synchronize dynamical systems, for examples, [11]-[13]. And impulsive control technique could be an efficient method when a discrete change behavior is needed. The adjustment interest rate could agree with that. In this paper, we proposed an impulsive

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 50–57, 2012.

synchronization scheme for a state delayed discrete complex networks with switching topology. For this control scheme, we consider that the impulsive control signal is designed to be input into all of nodes.

The paper is organized as follows. Section 2 presents some mathematical preliminaries needed in this work, and a generalized mathematical model for delayed discrete complex networks with switching topology. The main theorem for global synchronization of this type of discrete complex networks are then given in Section 3. In Section 4, a small-world networks with 3 sub-networks involving 30 nodes is constructed to illustrate the effectiveness of our result. Section 5 concludes the paper.

2 Preliminary

First, we need to introduce some notations and definitions for the sake of exploring our main results. Let$\| \bullet \|$ denote the Euclidean norm; $\mathbb{R}^n$ denotes the n–dimensional Euclidean space,the set of natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$, and, for certain positive integer τ, we let $\mathbb{Z}_{-\tau} = \{-\tau, -\tau+1, \ldots, 0\}$. The family of N linearly coupled discrete complex networks, consisting of time delay with respect to its system state and the switched topology, can be described by

$$\begin{aligned} x_i(k+1) = {} & Ax_i(k) + Bf(x_i(k)) + Df(x_i(k-\tau(k))) + I(k) \\ & + \sum_{j=1}^{N} c_{ij,\sigma(k)} \Gamma x_j(k-\tau(k)), \qquad i = 1, 2, ..., N, \qquad k \in \mathbb{N} \end{aligned} \tag{1}$$

$$x_{ik_0} = \phi(\theta), \qquad \theta \in \mathbb{Z}_{-\tau}, \tag{2}$$

where $x_i(k) = (x_{i,1}, x_{i,2}, ..., x_{i,n}) \in \mathbb{R}^n$ represents the state vector of the i–th node at every instant of time k and n denotes the number of nodes affiliated to each sub-networks. $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ are known real matrices. $\boldsymbol{f(x_i(k))} = (\boldsymbol{f_1(x_{i,1}(k)), f_2(x_{i,2}(k)), ..., f_n(x_{i,n}(k))})^T$ and $\boldsymbol{f(\bullet)} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is a smooth nonlinear vector-valued functions. $\boldsymbol{I(k)} = (I_i(k), I_2(k), ..., I_n(k))^T$ is a n-dimensional vector from external input. $\boldsymbol{S}$ is a finite index set of r elements: $\boldsymbol{S} = \{s_1, s_2, ..., s_r\}$. Let the switching function be denoted by $\sigma(k) : \mathbb{N} \longrightarrow \boldsymbol{S}$, which is the switching signal from sudden changing of system dynamic without jumps in the state $\boldsymbol{x}$ at any switching instant. Specifically, we consider that it is a piecewise constant function and continuous from the right, indicating certain active subsystem regime, at every instant of time k the index $\sigma(k) = s_k \in \boldsymbol{S}$; meanwhile, let the switching instants of σ be denoted by $k_{m,x} (m = 1, 2, ...)$ and let $k_{0,x} := 0$ (without chattering). $C_{s_k} = (c_{ij,s_k}) \in \mathbb{Z}^{N\times N}$ represents the outer coupling configuration symmetric matrix defined as follows: for each active subsystem regime s_k , if there is a connection from node j to node i $(j \neq i)$, then $c_{ij,s_k} = c_{ji,s_k} > 0$; otherwise $c_{ij,s_k} = c_{ji,s_k} = 0$. Assume that

$$c_{ii,s_k} = -\sum_{j=1, j\neq i}^{N} c_{ij,s_k} = -\sum_{j=1, j\neq i}^{N} c_{ji,s_k}, \qquad i \in N, \qquad s_k \in \boldsymbol{S}. \tag{3}$$

The notation represents $\Gamma \in \mathbb{R}^{n\times n}$ the diagonal inner coupling matrix between two connected nodes. $\tau(k)$ is a time-varying delay with respect to each instant of time k and satisfies $\tau(k) \in \mathbb{Z}_{-\tau}$. $\phi(\bullet) : \mathbb{Z}_{-\tau} \longrightarrow \mathbb{R}^{n\times N}$ is continuous everywhere except at a finite number of points. The norm of $\phi(\bullet)$ is defined by $\|\phi(\theta)\|_\tau = \sup_{\theta\in\mathbb{Z}_{-\tau}}\{\|\phi(\theta)\|\}$. We assume that at each active subsystem regime, the existence and uniqueness of a solution of system (1) for every initial condition and piecewise/continuous input can be guaranteed. In order to design an impulsive control scheme to synchronize system (1), we consider the evolutionary state is abruptly jumping at every impulsive instant of time k_u from its open-loop state , which can be formularized by

$$\Delta x_i(k_{m,u}) = J_u x_i^*(k_{m,u}), \qquad m = 1, 2, ...\mathbb{N} \tag{4}$$

where $x_i^*(k_{m,u})$ stands for the primal state at time instant $k_{m,u}$ without impulsive jump. As usual, every impulsive instant of time $k_{l,u}$ satisfies $0 = k_{0,u} < k_{1,u} < k_{2,u} < \cdots < k_{m,u} < k_{m+1,u} < \cdots$ and $\lim_{m\to\infty} k_{m,u} = \infty$;$J_u : \mathbb{R}^n \to \mathbb{R}^n (m = 1, 2, ...)$ represents the impulsive jump strength. Therefore, at every impulsive instant of time $k_{m,u}$, the coupled states $x_i(k) - x_j(k)$ between connected node i and j can be described by

$$x_i(k_{m,u}) - x_j(k_{m,u}) = x_i^*(k_{m,u}) - x_j^*(k_{m,u}) + J_u[x_i^*(k_{m,u}) - x_j^*(k_{m,u})]. \tag{5}$$

Intuitively, a family of impulsive controller can be designed as

$$U_i(k, x_i(k)) = \sum_{u=1}^{\infty} \delta(k - k_{m,u}) J_u(x_i^*(k_{m,u})), \qquad m = 1, 2, ...\mathbb{N}, \tag{6}$$

where $U_i(k, x_i(k))$ represents a class of impulsive controller at each instant of time $k_{m,u}$; $\delta(\bullet)$ denotes the Dirac discrete-time function.

Assumption 1. *For each nonlinear function* $\boldsymbol{f_i(\bullet)}$ *(i=1,2,...,n),suppose that it is globally Lipschitz continues function and satisfies*

$$\|f_i(x_1) - f_i(x_2)\| \le \hat{l}_i\|x_1 - x_2\|, \quad i = 1, 2, ..., n, for\, any\, x_1, x_2 \in \mathbb{R}, \tag{7}$$

where $\hat{l}_i$ *is certain positive constant.*

Definition 1. *The system of the impulsive controlled discrete complex networks (7) is said to be globally exponentially synchronized, if for any initial condition* $\phi(\bullet) : \mathbb{Z}_{-\tau} \to \mathbb{R}^{n\times N}$*, and there exist two positive constants* λ *and* $M_0 \ge 1$ *such that*

$$\|x_i(k) - x_j(k)\| \le M_0 e^{-\lambda(k-k_0)}, \quad 1 \le i \le j \le N \tag{8}$$

holds for all $k > k_0$*.*

Lemma 1. *Let* $\boldsymbol{W} = (w_{ij})_{N\times N}$*,* $\boldsymbol{P} \in \mathbb{R}^{n\times n}$*,*$\boldsymbol{x} = (x_1, x_2, ..., x_N)^T$ *and* $\boldsymbol{y} = (y_1, y_2, ..., y_N)^T$ *with* x_k*,*$y_k \in \mathbb{R}^n$ *(k=1,2,...,N). If* $\boldsymbol{W} = \boldsymbol{W}^T$ *and each row sum of* $\boldsymbol{W}$ *is zero, then*

$$\boldsymbol{x^T}(\boldsymbol{W} \otimes \boldsymbol{P})\boldsymbol{y} = -\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} w_{ij}(x_i - x_j)^T P(y_i - y_j). \tag{9}$$

3 Main Results

When the impulsive controller can be functioning simultaneously at the state of discrete complex networks' switching signal, the equivalent impulsive controlled system is rewritten by using the matrix Kronecker product

$$\begin{aligned} x(k+1) &= (I_N \otimes A)x(k) + (I_N \otimes B)F(x(k)) + (I_N \otimes D)F(x(k-\tau)) \\ &\quad + I(k) + (C_{\sigma(k)} \otimes \Gamma)x(k-\tau), \quad k \neq k_{m,u} \end{aligned} \tag{10}$$

$$x(k_{m,u}) = [I_N \otimes (I_N + J_u(k_{m,u}))]x(k_{m,u}-1), \tag{11}$$

for any k,$m \in \mathbb{N}$.

Theorem 2. *Under* ***Assumption 1.*** *the impulsive controlled complex network(12) is exponentially synchronized if there exists certain positive integer m_τ, positive scalars $\varepsilon_{\sigma(k)}$,$p_{\sigma(k)}$, $q_{\sigma(k)}$ and positive-definite matrices $P_{\sigma(k)} \in \mathbb{R}^{n\times n}$,$Q_{l,\sigma(k)} \in \mathbb{R}^{n\times n}$ (l=1,2,...6) such that*

(i) Given $\mu \geq 1$ and $P_{\sigma(k_{m,x})} \leq \mu P_{\sigma(k_{m+1,x})}$, for any $k \in [k_{m,x}, k_{m+1,x}-1]$ in corresponding sub-state $\sigma(k_{m,x})$,

$$p_{\sigma(k_{m,x})} - \left[\frac{\lambda_{max}(\Pi_{\sigma(k_{m,x})})}{\lambda_{min}(P^{-1}_{\sigma(k_{m,x})})} + \mu q_{\sigma(k_{m,x})}\frac{\lambda_{max}(\Omega_{\sigma(k_{m,x})})}{\lambda_{min}(P^{-1}_{\sigma(k_{m-m_\tau,x})})}\right] \geq 0, \tag{12}$$

where

$$\begin{aligned} \Pi_{\sigma(k_{m,x})} &= A^T P_{\sigma(k_{m,x})} A + L^T B^T P_{\sigma(k_{m,x})} BL + A^T Q_{1,\sigma(k_{m,x})} A \\ &\quad + L^T B^T P^T_{\sigma(k_{m,x})} Q_{1,\sigma(k_{m,x})} P_{\sigma(k_{m,x})} BL + A^T Q^{-1}_{2,\sigma(k_{m,x})} A \\ &\quad - NC_{\sigma(k_{m,x})} A^T Q^{-1}_{3,\sigma(k_{m,x})} ANC_{\sigma(k_{m,x})} + L^T B^T Q_{4,\sigma(k_{m,x})} BL \\ &\quad - L^T NC_{\sigma(k_{m,x})} B^T Q_{5,\sigma(k_{m,x})} BNC_{\sigma(k_{m,x})} L, \\ \Omega_{\sigma(k_{m,x})} &= L^T D^T P_{\sigma(k_{m,x})} DL - NC^2_{\sigma(k_{m,x})} \Gamma^T P_{\sigma(k_{m,x})} \Gamma \\ &\quad + L^T D^T P^T_{\sigma(k_{m,x})} Q_{2,\sigma(k_{m,x})} P_{\sigma(k_{m,x})} DL \\ &\quad - \Gamma^T P^T_{\sigma(k_{m,x})} Q_{3,\sigma(k_{m,x})} P_{\sigma(k_{m,x})} \Gamma \\ &\quad + L^T D^T P^T_{\sigma(k_{m,x})} Q_{4,\sigma(k_{m,x})} P_{\sigma(k_{m,x})} DL \\ &\quad - \Gamma^T P^T_{\sigma(k_{m,x})} Q_{5,\sigma(k_{m,x})} P_{\sigma(k_{m,x})} \Gamma \\ &\quad - L^T NC_{\sigma(k_{m,x})} D^T Q^{-1}_{6,\sigma(k_{m,x})} DNC_{\sigma(k_{m,x})} L \\ &\quad - \Gamma^T P^T_{\sigma(k_{m,x})} Q_{6,\sigma(k_{m,x})} P_{\sigma(k_{m,x})} \Gamma. \end{aligned}$$

(ii) $\mu\lambda^2_{max}(1 + J_u(k_{m,x})) < e^{\varepsilon_{\sigma(k_{m,x})}(k_{m+1,x}-k_{m,x})}$.

(iii) $q_{\sigma(k_{m,x})} \geq e^{\varepsilon_{\sigma(k_{m,x})}(k_{m+1,x}-k_{m,x}+1)+\sum_{i=0}^{m_\tau-1}\varepsilon_{k_{m-i,x}}(k_{m+1-i,x}-k_{m-i,x})}$, *where* $m_\tau = \lceil \frac{\tau}{\inf\{k_{m,x}-k_{m-1,x}\}} \rceil$.

Proof. Consider the following Lyapunov function:

$$V(k) = x^T(k)(W \otimes P_{\sigma(k)})x(k), \tag{13}$$

for any $k \in [k_{m,x}, k_{m+1,x} - 1]$, m=1,2,...
where

$$W = \begin{bmatrix} N-1 & -1 & ... & -1 \\ -1 & N-1 & ... & -1 \\ ... & ... & ... & ... \\ -1 & -1 & ... & N-1 \end{bmatrix}$$

.

One observes that for the case $k \in \mathbb{Z}_\tau$,

$$\begin{aligned} V(\theta) &= x^T(\theta)(W \otimes P_{\sigma(0)})x(\theta) \\ &= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (x_i(\theta) - x_j(\theta))^T P_{\sigma(0)}(x_i(\theta) - x_j(\theta)) \\ &= \varphi(\|\phi(\theta)\|_\tau^2). \end{aligned} \tag{14}$$

Choose $M \geq 1$, such that

$$\begin{aligned} \varphi(\|\phi(\theta)\|_\tau^2) &\leq M\varphi(\|\phi(\theta)\|_\tau^2)e^{-\lambda(k_{1,x}-k_{0,x})}e^{-\varepsilon_{\sigma(k_{0,x})}(k_{1,x}-k_{0,x})} \\ &< q_{\sigma(k_{0,x})}\varphi(\|\phi(\theta)\|_\tau^2). \end{aligned} \tag{15}$$

By claiming that

$$V(k) \leq M\varphi(\|\phi(\theta)\|_\tau^2)e^{-\lambda(k_{m,x}-k_{0,x})}, k \in [k_{m-1,x}, k_{m,x} - 1],\ m \in N. \tag{16}$$

And by virtue of mathematical induction, the claim (16) is true for each $k \in \mathbb{N}$. In view of (16) and **Definition 1.**, it can be obtained that

$$V(k) \leq M\varphi(\|\phi(\theta)\|_\tau^2)e^{-\lambda(k-k_{0,x})},\ k \in [k_{m-1,x}, k_{m,x} - 1],\ m \in \mathbb{N}. \tag{17}$$

For any $k \in \mathbb{N}$,

$$\begin{aligned} &\min\left\{\lambda_{\min}(P_{\sigma(k)})\right\} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \|x_i(k) - x_j(k)\|^2 \\ &\leq \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (x_i(k) - x_j(k))^T P_{\sigma(k)}(x_i(k) - x_j(k)) \\ &\leq M\varphi(\|\phi(\theta)\|_\tau^2)e^{-\lambda(k-k_{0,x})}. \end{aligned} \tag{18}$$

Therefor, for any $k \in \mathbb{N}$,

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \|x_i(k) - x_j(k)\|^2 \leq \min\left\{\lambda_{\min}^{-1}(P_{\sigma(k)})\right\} M\varphi(\|\phi(\theta)\|_\tau^2)e^{-\lambda(k-k_{0,x})}, \tag{19}$$

which implies

$$\|x_i(k) - x_j(k)\| \leq M_0 e^{-\lambda(k-k_{0,x})}, \qquad 1 \leq i \leq j \leq N. \tag{20}$$

Therefore, the discrete complex networks (1) is globally exponentially synchronized under impulsive control. The proof is thus completed. □

Remark 1. We consider a multiple Lyapunov function for each sub-networks with arbitrarily fast switching signal in our theorem, which results in a less conservation criterion.

Remark 2. In the switched Lyapunov function, $p_{\sigma(k)}$ gives an upper bound on the estimation of divergence rate for each running sub-networks. By condition (ii) of **Theorem 1.**, the impulsive control gain is designed to compensate divergence from system itself and deteriorating effect from arbitrarily fast switching. If some certain sub-networks could be self-synchronizing, the impulsive control gain only need to compensate deteriorating effect.

4 Example and Numerical Simulations

This section presents a typical example to illustrate our result. Let us consider a 2-dimensional discrete chaotic neural networks is given as the isolated node of a small world network with 30 nodes,

$$x(k+1) = Ax(k) + Bf(x(k)) + Df(x(k-\tau(k))) + I(k), \tag{21}$$

where $x(k) = (x_1(k), x_2(k))^T$,$f(x(k)) = (tanh(x_1(k)), tanh(x_2(k)))^T$,$I(k) = (0,0)^(T)$,

$$A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, B = \begin{bmatrix} 2 & -0.11 \\ -5 & 3.2 \end{bmatrix}, D = \begin{bmatrix} -1.6 & -0.1 \\ -0.18 & -2.4 \end{bmatrix},$$

and $\tau(k) = \frac{e^k}{1+e^(k)}$.Obviously, Lipschitz constants can be 1 here.Consider a small-world model involved with three different subsystem. The trajectory of each single node of this small-world model has random initial values in the interval [0.3,3] and [-3,-0.3], respectively. Given a switching signal $\sigma(t)$ in Fig1.(a), we have the state response of the switched complex networks, see Fig.1(b). From **Theorem 1**, for each sub-network, we have $J_1 = \begin{bmatrix} -0.6667 & 0 \\ 0 & -0.667 \end{bmatrix}$. $J_2 = \begin{bmatrix} -0.4079 & 0 \\ 0 & -0.4079 \end{bmatrix}$. $J_3 = \begin{bmatrix} -1.1576 & 0 \\ 0 & -1.1576 \end{bmatrix}$.

It is shown that all of nodes in each sub-networks could not reach into a synchronous state without a control. Indeed, the switched signal plays a role of deterioration accelerator to diverge the synchronous state,shown in Fig.1(b). Once the feasible impulsive controller is placed on discrete complex networks with topology switching, such complex networks would be synchronized, see Fig.1(c).

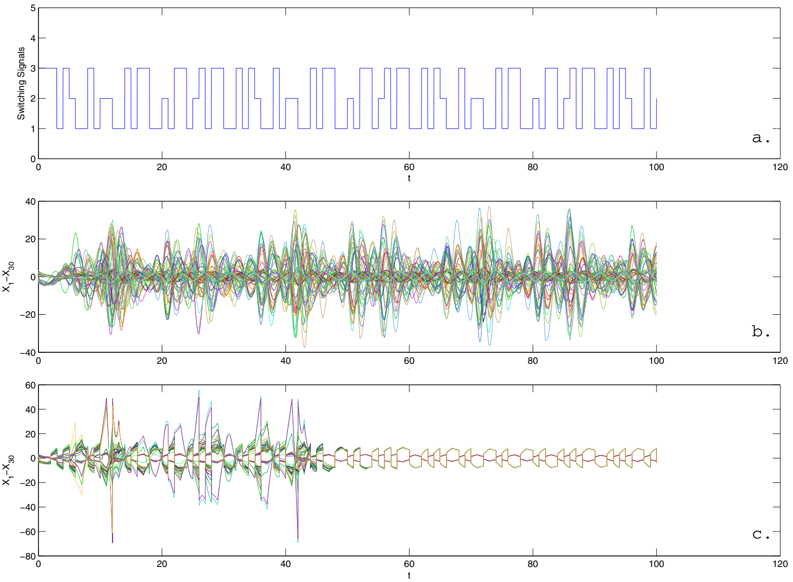


Fig. 1. (a) The switching signal σ(t); (b)The state responses of the switched system; (c)The synchronized state under impulsive control

5 Conclusion

In this paper, we have investigated impulsive synchronization control of a discrete delayed complex networks with switching topology by using Lyapunov Ruzimiki method. A time-varying delay dependent criteria for exponential synchronization is presented guarantee the switched discrete complex networks tending to be a synchronous manifold. It is worthwhile to see time-varying delay can take any value, even larger than any dwell time of a sub-networks.Futhermore, a numerical example with 3 sub-networks are presented by using the impulsive control technique.

Acknowledgements. This research is supported by Australia Government grant through the Collaborative Research Network(CRN) to the university of Ballarat.

References

1. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393, 440–442 (1998)
2. Barab, A.L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
3. Strogatz, S.H.: Exploring Complex Networks. Nature 410, 268–276 (2001)
4. VanWiggeren, G.D., Roy, R.: Communication with chaotic lasers. Science 279, 1198–1200 (1998)

5. Fischer, I., Liu, Y., Davis, P.: Synchronization of chaotic semiconductor laser dynamics on subnanosecond time scales and its potential for chaos communication. Physical Review A 62, 011801-1–011801-4 (2000)
6. Hoppensteadt, F.C., Izhikevich, E.M.: Pattern recognition via synchronization in phase-locked loop neural networks. IEEE Transacations on Neural Networks 11, 734–738 (2000)
7. Wang, X.F., Chen, G.R.: Synchronization in scale-free dynamical networks:robustness and fragility. IEEE Transacations on Circuits and Systems – I, Reg. Papers. 49, 54–62 (2002)
8. Wu, C.: Synchronization in networks of nonlinear dynamical systems coupled via a directed graph. Nonlinearity 18, 1057–1064 (2005)
9. Wang, Z., Wang, Y., Liu, Y.: Global synchronization for discrete-time stochastic complex networks with randomly occurred nonlinearities and mixed time delays. IEEE Transacations on Neural Networks 21, 11–25 (2010)
10. Li, X., Wang, X.F., Chen, G.: Pinning a complex network to its equilibrium. IEEE Transacations on Circuits and Systems – I, Reg. Papers 51, 2074–2087 (2004)
11. Li, C.D., Shen, Y.Y., Feng, G.: Stabilizing effects of impulse in delayed BAM neural networks. IEEE Transactions on Circuits and Systems–II, Brief papers 53, 1284–1288 (2008)
12. Li, C.J., Li, C.D., Liao, X.F., Huang, T.W.: Impulsive effects on stability of high-order BAM neural networks with time delays. Neurocomputing 74, 1541–1550 (2011)
13. Li, C.J., Li, C.D., Huang, T.W.: Exponential stability of impulsive high order Hopfield-type neural networks with delays and reaction–diffusion. International Journal of Computer Mathematics 88, 3150–3162 (2011)
14. Lu, J., Ho, D.W.C., Wu, L.: Exponential stabilization in switched stochastic dynamical networks. Nonlinearity 22, 889–911 (2009)
15. Huang, T., Chen, G., Kurth, J.: Synchronization of chaotic systems with time-varying coupling delays. Discrete and Continuous Dynamical Systems – Series B 16, 1071–1082 (2011)
16. Huang, T., Li, C., Gao, D., Xiao, M.: Anticipating synchronization through optimal feedback control. Journal of Global Optimization 52, 281–290 (2012)

Salient Instance Selection for Multiple-Instance Learning

Liming Yuan*, Songbo Liu, Qingcheng Huang, Jiafeng Liu, and Xianglong Tang

School of Computer Science and Technology, Harbin Institute of Technology,
Harbin 150001, China
yuanleeming@163.com,
{sbliu,huangqc,jefferyliu,tangxl}@hit.edu.cn

Abstract. Multiple-instance learning (MIL) is a variant of traditional supervised learning, where training examples are bags of instances. In this learning framework, only the labels of bags are known while the labels of instances in bags are unknown. This ambiguity in labels of instances leads to significant challenges in MIL. In this paper, we propose an efficient instance selection method to solve this problem, called Salient Instance Selection for Multiple-Instance Learning (MILSIS). MILSIS has two roles: first, selecting discriminative instances and eliminating redundant or irrelevant instances from each bag; second, selecting an instance prototype from each positive bag to construct an embedding space in order to convert the MIL problem to the standard single instance learning problem. Accordingly, based on the first role, we present two novel MIL methods, called MILSIS-kNN-C and MILSIS-kNN-B; based on the second role, we present another new MIL method, called MILSIS-SVM. Experimental results on some synthetic and benchmark data-sets demonstrate the effectiveness of our methods as compared to others.

Keywords: Multiple-instance Learning, Salience, Instance Selection, K-nearest Neighbor Classification, Support Vector Machines.

1 Introduction

Multiple-instance learning (MIL) was first introduced by Dietterich et al. when they were investigating the problem of drug activity prediction [1]. In this learning framework, each training example is a bag composed of one or more instances. A bag is positive if it contains at least one positive instance; otherwise, negative. The labels of training bags are known while the labels of instances in those bags are unknown. Since then, MIL has gained significant attention in the machine learning and computer vision communities [2–6].

Many MIL methods have been proposed during the past decade. To name a few, APR [1], DD [7], Multi Instance Neural Networks (MI-NN) [8], Citation-kNN and Bayesian-kNN [9], EM-DD [10], MI-SVM and mi-SVM [11]. Recently,

* Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 58–67, 2012.

several instance selection-based methods have been proposed, namely DD-SVM [12], MILES [13], MILD [14] and MILIS [15], which tackle the MIL problem by converting MIL into single instance learning (SIL). The basic idea is mapping each bag into a new feature space constituted by some instance prototypes (IP) selected from training bags, and then learning a SVM classifier in this new feature space named the embedding space. It should be noted here that the main difference between these methods is how to select IPs, and a good instance selection method may lead to a good performance.

In this paper, we propose a new instance selection method to tackle the MIL problem, named Salient Instance Selection for Multiple-Instance Learning (MILSIS). The novelty lies in considering the salience of every instance in each positive bag, which will be detailed in Section 2. Then we present three MIL methods based on MILSIS in Section 3, named MILSIS-kNN-C, MILSIS-kNN-B and MILSIS-SVM. In Section 4, we evaluate and discuss our methods on some synthetic and benchmark data-sets. In Section 5, we conclude and give some future research directions.

2 Salient Instance Selection for MIL

2.1 Notations

Let B_i^+ denote a positive bag and B_i^- denote a negative bag. Accordingly, B_{ij}^+ denotes an instance in B_i^+ and B_{ij}^- denotes an instance in B_i^-. Let $\mathcal{B} = \{B_1^+, B_2^+, \ldots, B_{n^+}^+, B_1^-, B_2^-, \ldots, B_{n^-}^-\}$ denote a training set comprised of n^+ positive bags and n^- negative bags. For the sake of simplicity, we will denote a bag as B_i with B_{ij}(s) when the label of it does not matter. Without ambiguity, B_{ij} also represents the feature vector of it depending on the context. $l(B_i)$ and $l(B_{ij})$ are the labels associated with B_i and B_{ij}, respectively. Note that $l(B_{ij})$ is not directly observable.

2.2 MILSIS

From the definition of MIL, we know that the ambiguity in labels arises from positive bags, since there may be not only positive instances but also negative instances in a positive bag, whereas all instances in each negative bag are labeled as negative. Our MILSIS method is aiming at identifying *true* positive instances in positive bags.

Assumption 1. *In general, any two positive or negative instances are close to each other while any positive and negative instance are far from each other.*

Definition 1. *$\forall B_{ij} \in B_i$, the salience of B_{ij} is defined as follows:*

$$Sal(B_{ij}) = \sum_{B_{ik} \in B_i \setminus \{B_{ij}\}} d(B_{ij}, B_{ik}) \ , \qquad (1)$$

where $d(\cdot,\cdot)$ is a distance function between two instances, which takes the form of Euclidean distance here.

Remark 1. Definition 1 indicates that an instance would be further from all other instances in the same bag if its salience is higher; otherwise, it would be closer to them.

Theorem 1. *Assume that $B_{ij} \in B_i^+$ is the only one positive or negative instance. Then*

$$\forall B_{ik} \in B_i^+ \setminus \{B_{ij}\}, \quad Sal(B_{ij}) > Sal(B_{ik}) \ . \tag{2}$$

Proof. By Definition 1, the salience of B_{ij} and B_{ik} is

$$Sal(B_{ij}) = \sum_{B_{it} \in B_i^+ \setminus \{B_{ij}\}} d(B_{ij}, B_{it}) = d(B_{ij}, B_{ik}) + \sum_{B_{it} \in B_i^+ \setminus \{B_{ij}, B_{ik}\}} d(B_{ij}, B_{it}) \ , \tag{3}$$

$$Sal(B_{ik}) = \sum_{B_{it} \in B_i^+ \setminus \{B_{ik}\}} d(B_{ik}, B_{it}) = d(B_{ik}, B_{ij}) + \sum_{B_{it} \in B_i^+ \setminus \{B_{ik}, B_{ij}\}} d(B_{ik}, B_{it}) \ . \tag{4}$$

Hence:

$$Sal(B_{ij}) - Sal(B_{ik}) = \sum_{B_{it} \in B_i^+ \setminus \{B_{ij}, B_{ik}\}} [d(B_{ij}, B_{it}) - d(B_{ik}, B_{it})] \ , \tag{5}$$

where $d(B_{ij}, B_{it})$ is the distance between a positive and negative instance, while $d(B_{ik}, B_{it})$ is the distance between two negative or positive instances. Therefore, we would have $Sal(B_{ij}) > Sal(B_{ik})$ by Assumption 1. □

Remark 2. We now consider the general case of Theorem 1, i.e. there are multiple positive and negative instances in a positive bag. Let m^+ and m^- denote the numbers of positive and negative instances, respectively. Then the salience of any positive instance largely depends on m^- distances between the positive instance and other negative instances by Definition 1 and Assumption 1 while the salience of any negative instance largely depends on m^+ distances between the negative instance and other positive instances. *Assuming* that the distance between any positive and negative instance fluctuates near a fixed value, then the above salience is largely dependent on m^- and m^+. Thus, the salience of any positive instance would be greater than that of any negative instance if m^- is greater than m^+; otherwise, the former would be less than the latter. Strictly speaking, the above assumption does not always hold since there may exist outliers and noise in the data-set. However, it might hold within the small scope of a single bag and could be largely satisfied according to the better experimental results in Section 4, at least on data-sets used in this paper. We give the formal definition of the above description in Generalization 1.

Generalization 1. *Assume that $\{B_{i1}^+, B_{i2}^+, \ldots, B_{im^+}^+\} \subset B_i^+$ is the subset of positive instances and $\{B_{i1}^-, B_{i2}^-, \ldots, B_{im^-}^-\} \subset B_i^+$ is the subset of negative instances. Then, $\forall j \in \{1, 2, \ldots, m^+\}$, $k \in \{1, 2, \ldots, m^-\}$,*

$$Sal(B_{ij}^+) \begin{cases} > Sal(B_{ik}^-) & \mathit{if}\ m^+ < m^- \ , \\ < Sal(B_{ik}^-) & \mathit{if}\ m^+ > m^- \ . \end{cases} \tag{6}$$

If we can estimate which of m^+ and m^- is greater given a positive bag, we could depend on Generalization 1 to select *true* positive instances from the positive bag. When $m^+ < m^-$, the salience of any positive instance is greater than that of any negative instance by Generalization 1. Note that negative instances dominate the positive bag in this case. Thus, the higher its salience, the further a candidate positive instance from all other negative instances by Definition 1 and Assumption 1, which means it is more likely to be positive. In this case, we can select instances with high salience as *true* positive ones. Similarly, when $m^+ > m^-$, we can select instances with low salience as *true* positive ones. The following Theorem 2 offers one possible solution to the above problem.

Definition 2. *Let* $\mathcal{B}^- = \{B_{rt} | B_{rt} \in B_r^-, r = 1, 2, \ldots, n^-\}$ *be the given set of negative instances. The probability that an instance* B_{ij} *is positive given* $\mathcal{B}^-$ *is:*

$$\Pr(l(B_{ij}) = 1 | \mathcal{B}^-) = 1 - \exp(-D(B_{ij}, \mathcal{B}^-)/\sigma^2) \ , \tag{7}$$

where σ *is a scaling factor larger than 0, and*

$$D(B_{ij}, \mathcal{B}^-) = \min_{B_{rt} \in \mathcal{B}^-} d(B_{ij}, B_{rt}) \ . \tag{8}$$

Remark 3. From Definition 2, we can easily deduce that $0 \leq \Pr(l(B_{ij}) = 1|\mathcal{B}^-) \leq 1$, $\Pr(l(B_{ij}) = 1|\mathcal{B}^-) = 0$ when $D(B_{ij}, \mathcal{B}^-) = 0$ and $\Pr(l(B_{ij}) = 1|\mathcal{B}^-) = 1$ when $D(B_{ij}, \mathcal{B}^-) = +\infty$. This is well consistent with our intuition. If an instance is far from a set of negative instances, they would have a low similarity and hence the instance is likely to be labeled as positive; otherwise, the instance is likely to be labeled as negative.

Theorem 2. $\forall B_i^+$*,* $\tilde{B}_i^+$ *is its corresponding re-sorted bag in descending order of the salience of instances. Let* m^+ *and* m^- *be the numbers of positive and negative instances in* $\tilde{B}_i^+$*, respectively, and* $m = m^+ + m^-$*. Assume that* $\mathcal{B}^-$ *is the given set of negative instances, then*

$$m^+ \begin{cases} < m^- & if \ \ \Pr(l(\tilde{B}_{i1}) = 1|\mathcal{B}^-) > \Pr(l(\tilde{B}_{im}) = 1|\mathcal{B}^-) \ , \\ > m^- & if \ \ \Pr(l(\tilde{B}_{i1}) = 1|\mathcal{B}^-) < \Pr(l(\tilde{B}_{im}) = 1|\mathcal{B}^-) \ . \end{cases} \tag{9}$$

Proof. Premise 1: $\Pr(l(\tilde{B}_{i1}) = 1|\mathcal{B}^-) > \Pr(l(\tilde{B}_{im}) = 1|\mathcal{B}^-)$. Assume $m^+ > m^-$, then we would have by Generalization 1:

$$Sal(\tilde{B}_{ij}^+) < Sal(\tilde{B}_{ik}^-) \quad \forall j \in \{1, 2, \ldots, m^+\}, k \in \{1, 2, \ldots, m^-\} \ . \tag{10}$$

And $Sal(\tilde{B}_{i1}) > Sal(\tilde{B}_{im})$, thus $\tilde{B}_{i1}$ is a negative instance and $\tilde{B}_{im}$ is a positive instance. Now by Assumption 1, $D(\tilde{B}_{i1}, \mathcal{B}^-) < D(\tilde{B}_{im}, \mathcal{B}^-)$. Then we will have, by (7):

$$\Pr(l(\tilde{B}_{i1}) = 1|\mathcal{B}^-) < \Pr(l(\tilde{B}_{im}) = 1|\mathcal{B}^-) \ . \tag{11}$$

This would contradict *Premise* 1, and thus cannot happen. So $m^+ < m^-$.
Premise 2: $\Pr(l(\tilde{B}_{i1}) = 1|\mathcal{B}^-) < \Pr(l(\tilde{B}_{im}) = 1|\mathcal{B}^-)$. Similarly, one can prove that $m^+ > m^-$ in *Premise* 2. □

Algorithm 1. MILSIS

```
Input: Training set $\mathcal{B}$, the number of salient instances per bag $SalNum$
Output: the set of IPs $T$
$\mathcal{B}^- = \{B_{rt} | B_{rt} \in B_r^-, r = 1, 2, \ldots, n^-\}$
$optPosInst$ = RoughSelection($\mathcal{B}, \mathcal{B}^-$)
$T$ = FineSelection($\mathcal{B}, \mathcal{B}^-, optPosInst, SalNum$)
```

Procedure 1. RoughSelection

```
Input: Training set $\mathcal{B}$, the set of negative instances $\mathcal{B}^-$
Output: the optimal positive instance $optPosInst$
$maxDist = 0$
for $i = 1$ to $n^+$ do
    Compute $Sal(B_{ij}^+)$ for each instance in $B_i^+$ by Definition 1
    Re-sort all instances in $B_i^+$ in descending order of salience
    Compute $D(B_{i1}^+, \mathcal{B}^-)$ and $D(B_{im}^+, \mathcal{B}^-)$ by (8) // $m$ is the number of instances
    if $D(B_{i1}^+, \mathcal{B}^-) > D(B_{im}^+, \mathcal{B}^-)$ and $D(B_{i1}^+, \mathcal{B}^-) > maxDist$ then
        $maxDist = D(B_{i1}^+, \mathcal{B}^-)$ and $optPosInst = B_{i1}^+$
    else if $D(B_{im}^+, \mathcal{B}^-) > D(B_{i1}^+, \mathcal{B}^-)$ and $D(B_{im}^+, \mathcal{B}^-) > maxDist$ then
        $maxDist = D(B_{im}^+, \mathcal{B}^-)$ and $optPosInst = B_{im}^+$
return $optPostInst$
```

Procedure 2. FineSelection

```
Input: Training set $\mathcal{B}$, the set of negative instances $\mathcal{B}^-$, the optimal positive instance
    $optPosInst$, the number of salient instances per bag $SalNum$
Output: the set of IPs $T$
$optNegInst = \arg\max_{t \in \mathcal{B}^-} d(t, optPosInst)$
for $i = 1$ to $n^+$ do
    if $d(B_{i1}^+, optNegInst) > d(B_{im}^+, optNegInst)$ then
        Add $B_{i1}^+, \ldots, B_{iSalNum}^+$ to $T$
    else
        Add $B_{i(m-SalNum+1)}^+, \ldots, B_{im}^+$ to $T$
return $T$
```

Since $\Pr(l(B_{ij}) = 1 | \mathcal{B}^-)$ is proportional to $D(B_{ij}, \mathcal{B}^-)$ in (7), we only need to compute $D(\tilde{B}_{i1}, \mathcal{B}^-)$ and $D(\tilde{B}_{im}, \mathcal{B}^-)$ for the comparison in (9).

Based on the above foundations, we can now present our MILSIS method. MILSIS is composed of two procedures, named "Rough Selection" and "Fine Selection". In "Rough Selection", we consider **instances in all negative bags** as $\mathcal{B}^-$ in Theorem 2 and select one *true* positive instance from each positive bag according to Generalization 1, then choose **an optimal positive instance** out of them which is furthest from $\mathcal{B}^-$. In "Fine Selection", we first select **an optimal negative instance** from $\mathcal{B}^-$, which is furthest from **the optimal positive instance**. Then we regard it as $\mathcal{B}^-$ and select *true* positive instances again. Algorithm 1 summarizes the whole process.

MILSIS has two major advantages: first, unlike the instance selection method in DD-SVM [12], MILSIS relies on the optimal negative instance to select IPs, so it is more robust to labeling noise; second, the computational cost of MILSIS is from two aspects, i.e. inside each positive bag and between each positive bag and the set of all negative instances. Generally, the number of instances in any bag is much less than that of all negative instances, so the computational cost mainly depends on the latter. Assuming the number of all positive bags is n^+, the number of instances in all positive bags is v^+ and the number of instances in all negative bags is v^-, then the computational cost of MILSIS is approximately $O(2n^+v^-) + O(2n^+)(\approx O(n^+v^-))$ according to Algorithm 1. As for the instance selection methods in DD-SVM [12], MILES [13], MILD [14] and MILIS [15], the computational costs are $O((v^+ + v^-)(v^+ + v^-))$, $O((v^+ + v^-)(v^+ + v^-))$, $O(v^+(v^+ + v^-))$ and $O(v^+v^-)$, respectively. In general, n^+ is less than v^+ unless the number of instances per positive bag equals to 1, which is virtually impossible since MIL has become SIL in this case. Therefore, MILSIS has the lowest computational cost among all these instance selection methods.

3 Multiple-Instance Learning Methods Based on MILSIS

3.1 MILSIS-kNN-C and MILSIS-kNN-B

In “Fine Selection” of Algorithm 1, we use the selected IPs from each positive bag to substitute the original bag. As for the negative bags, we deal with them in the same way. There are two reasons for dealing with the negative bags like this: first, the selected instances are still negative ones; second, since we have to use the same scheme to handle *testing* bags, the instances far from the optimal negative instance will be selected from every negative *testing* bag. If we select instances near to the optimal negative instance from negative *training* bags, the substitute for each negative *testing* bag is obviously likely to be wrongly classified since the substitutes for negative *training* bags are far from it; otherwise, the probability that it is correctly classified will be high. After the above procedure is finished, we use the substitute bags to learn Citation-kNN or Bayesian-kNN [9]. Note that when the number of instances in a bag is less than that of IPs per bag ($SalNum$, refer to Algorithm 1), we select all instances from this bag and hence different substitute bags may have different numbers of instances.

3.2 MILSIS-SVM

We first employ MILSIS to select an IP from each positive bag and use all the selected IPs to construct the corresponding embedding space. Next, we map each bag to a point in this embedding space with the feature mapping function defined in (12). Finally, we use these new feature vectors to learn a standard SVM classifier with the RBF kernel.

Definition 3. *The feature mapping function of* MILSIS-SVM *is defined as*

$$f(\cdot) = [D(\cdot, t_1), \ldots, D(\cdot, t_i), \ldots, D(\cdot, t_{n^+})] \ , \tag{12}$$

where $t_i \in T$*,* $i = 1, 2, \ldots, n^+$*,* T *is the set of* IPs *and* n^+ *is the size of* T*.*

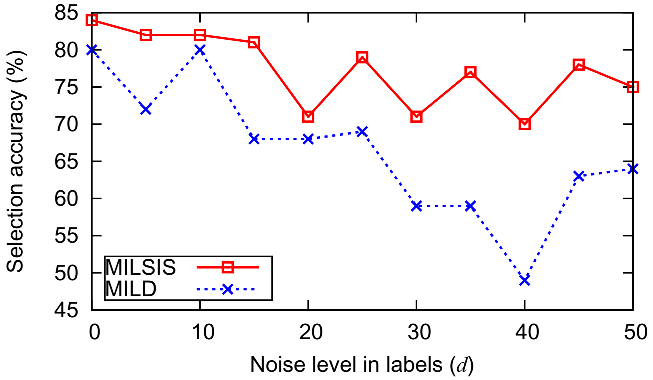


Fig. 1. Robustness of MILSIS and MILD to labeling noise

4 Experiments and Discussion

4.1 MILSIS on Synthetic Data-Sets

We use the synthetic data-sets generated out of the MNIST database[1] of handwritten digits to evaluate the performance of MILSIS. Since the digits 0 and 6 are similar in appearance, they are considered as the positive and negative instances, respectively. We randomly generate 100 positive bags and 100 negative bags, each containing 5 instances. Every positive bag contains only one positive instance. To evaluate the robustness of MILSIS to labeling noise, we randomly generate $d\%$ of positive bags to substitute $d\%$ of random negative bags, then deliberately mislabel these positive bags as negative to create noise in labels. This process is performed ten times for a single noise level $d\%$ and the average selection accuracies of MILSIS and MILD [14] on different noise levels of data-sets are shown in Fig. 1. Here MILD is chosen for comparison since the instance selection method in it also tries to uncover the properties of *true* positive instances in each positive bag as our MILSIS method does. Obviously, MILSIS is superior to MILD with respect to performance and robustness. Moreover, the computation time of MILSIS is only 0.8 second on a a 3GHz PC with 8GB memory while that of MILD is 7.7 seconds, thus MILSIS is more efficient than MILD.

4.2 MILSIS-Based MIL Methods on Benchmark Data-Sets

In the second experiment, we use five standard MIL benchmark data-sets, i.e. Musk1, Musk2, Elephant, Fox and Tiger[2], to evaluate our MILSIS-based methods. MILSIS-kNN-C and MILSIS-kNN-B have three parameters, i.e. $SalNum$ (for details, refer to Algorithm 1), $RefNum$ and $CiterRank$ [9], where

[1] The MNIST database is available at http://yann.lecun.com/exdb/mnist/

[2] These data-sets are available at
http://www.uco.es/grupos/kdis/mil/fs/#experiments/

$CiterRank$ equals to 0 for MILSIS-kNN-B. MILSIS-SVM has two parameters, i.e. the penalty parameter C and the kernel parameter γ, where LIBSVM [16] is applied to train all SVMs with the RBF kernel. In our experiments, $SalNum$ is restricted in $\{1, 2, \ldots, avgInstNum\}$, where $avgInstNum$ represents the average number of instances per bag. $RefNum$ is chosen from $\{1, 2, \ldots, 10\}$ and $CiterRank$ is set to $RefNum + 2$ based on the suggestion in [9]. Both C and γ are chosen from $\{2^{-10}, 2^{-9}, \ldots, 2^{9}, 2^{10}\}$. Those giving the maximum 2-fold cross-validation accuracy on each training set are chosen and fixed in the subsequent experiments. Note that $SalNum$ in MILSIS-kNN-B is not the optimal one but empirically set to that in MILSIS-kNN-C and $CiterRank$ predefined as $RefNum + 2$ may not be the best choice.

Then we randomly run 10 times of 10-fold cross-validation on each data-set and report the average classification accuracies and corresponding 95% confidence intervals in Table 1 where the best performance is highlighted in boldface. We also list some other results which are taken from [3, 14, 17]. We can see that MILSIS-kNN-C and MILSIS-kNN-B outperform Citation-kNN and Bayesian-kNN, respectively. MILSIS-SVM is competitive with the state-of-the-art MIL methods and better than them with respect to the average performance. The better performance is due to our MILSIS method, and this also validates that MILSIS could give a possible solution to the problem presented in [9], i.e. "how to remove those *false* positive instances from the positive bags".

Table 1 also shows that MILSIS-kNN-B is highly comparable to MILSIS-kNN-C, which indicates MILSIS could keep the effectiveness of kNN without resorting to *Citation* [9]. This is beneficial to the learning process since not *Reference* but *Citation* leads to most of the time consumption in Citation-kNN. Meanwhile, the computational complexity brought by MILSIS is very low according to Section 2.2 and Section 4.1. Therefore, we could use MILSIS to substitute *Citation* in the MIL setting. The computation time of 2-fold cross-validation on each dataset is given in Table 2. We can find that MILSIS-kNN-B is the most efficient one among all these methods. In addition, MILSIS-kNN-C and MILSIS-kNN-B are more efficient than Citation-kNN and Bayesian-kNN, respectively, which is mainly due to our MILSIS method. More importantly, the classification response is significantly speeded up with the help of MILSIS. Overall, MILSIS-kNN-B is the best one among all these kNN-based methods in terms of both classification accuracy and computation time.

For those instance selection-based methods in Table 1, the computation time is the sum of that spent on model selection and classifier learning. For all these methods, the computation time spent on classifier learning is close except MILIS [15] that employs a time-consuming iterative optimisation framework, while that spent on model selection mainly depends on the corresponding instance selection methods. Thus, according to the theoretical and empirical analysis in Section 2.2 and Section 4.1, we can conclude that MILSIS-SVM is the most efficient one among all these instance selection-based methods in Table 2.

Table 1. Classification accuracies (%) of various MIL methods on benchmark data-sets

Method	Musk1	Musk2	Elephant	Fox	Tiger	Avg.
MILSIS-kNN-C	90.9[±4.0]	**84.4**[±2.9]	**84.5**[±2.5]	63.5[±3.6]	**80.3**[±2.2]	**80.7**
MILSIS-kNN-B	**92.0**[±1.5]	**84.4**[±3.6]	81.7[±2.6]	**64.5**[±2.7]	78.5[±2.1]	80.2
Citation-kNN [9]	88.8[±4.7]	82.6[±4.0]	80.7[±2.5]	58.1[±2.8]	79.5[±3.1]	77.9
Bayesian-kNN [9]	87.8[±4.4]	80.6[±3.3]	73.3[±3.5]	59.5[±3.7]	77.7[±2.1]	75.8
MILSIS-SVM	**90.1**[±2.9]	85.6[±2.6]	81.8[±1.8]	**66.4**[±3.5]	80.8[±3.2]	**80.9**
MI-SVM [11]	77.9	84.3	81.4	57.8	**84.0**	77.1
mi-SVM [11]	87.4	83.6	82.2	58.2	78.4	78.0
DD-SVM [12]	85.8	**91.3**	83.5	56.6	77.2	79.0
MILES [13]	86.3	87.7	**84.1**	63.0	80.7	80.4
MILD_B [14]	88.3	86.8	82.9	55.0	75.8	77.8
MILIS [15]	88.6	91.1	N/A	N/A	N/A	N/A

Table 2. Computation time (in seconds) of various kNN-based MIL methods on benchmark data-sets (time spent on instance selection + time spent on classification.)

Data-set	MILSIS-kNN-C	MILSIS-kNN-B	Citation-kNN	Bayesian-kNN
Musk1	0.17 + 0.57	**0.17 + 0.36**	0 + 0.99	0 + 0.64
Musk2	16.64 + 5.52	**16.61 + 3.68**	0 + 162.81	0 + 80.95
Elephant	0.62 + 5.35	**0.63 + 3.47**	0 + 8.65	0 + 5.59
Fox	0.63 + 3.18	**0.63 + 2.02**	0 + 7.54	0 + 4.90
Tiger	0.60 + 4.19	**0.60 + 2.59**	0 + 6.53	0 + 4.26

5 Conclusions and Future Work

The intrinsic property making MIL difficult to tackle is that instance labels in positive bags are unknown, which is the essential difference between MIL and SIL. In this paper, we try to address this issue with an explicit instance selection method. Our method tries to uncover the characteristics of *true* positive instances in positive bags. This may be more valuable than simply making an accurate prediction since it can help us to understand the connections between instances and bags or the source of ambiguity in instance labels. Based on our instance selection method, we propose three novel MIL methods. Theoretical analysis and experimental results demonstrate that all our methods are more effective and efficient than the state-of-the-art.

We are considering several possible directions for our future work. First, more experiments would be done to validate our methods, especially large-scale datasets. Next, it is interesting to explore the scheme for noise removal since there exists labeling noise in most real applications. Finally, we are trying to extend our instance selection method to the multi-instance multi-label learning context.

Acknowledgments. This research has been supported by the National Natural Science Foundation of China under the Grant Nos. 61173087 and 61073128.

References

1. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the Multiple Instance Problem with Axis-Parallel Rectangles. Artif. Intell. 89, 31–71 (1997)
2. Viola, P., Platt, J., Zhang, C.: Multiple Instance Boosting for Object Detection. In: 18th International Conference on Advances in Neural Information Processing Systems, pp. 1417–1424. MIT Press, Cambridge (2006)
3. Fung, G., Dundar, M., Krishnapuram, B., Rao, B.R.: Multiple Instance Learning for Computer Aided Diagnosis. In: 19th International Conference on Advances in Neural Information Processing Systems, pp. 425–432. MIT Press, Cambridge (2007)
4. Babenko, B., Yang, M.H., Belongie, S.: Robust Object Tracking with Online Multiple Instance Learning. IEEE Trans. Pattern Anal. Mach. Intell. 33, 1619–1632 (2011)
5. Zhang, Q., Goldman, S.A., Yu, W., Fritts, J.E.: Content-Based Image Retrieval Using Multiple-Instance Learning. In: 19th International Conference on Machine Learning, pp. 682–689. Morgan Kaufmann, San Francisco (2002)
6. Rahmani, R., Goldman, S.A., Zhang, H., Cholleti, S.R., Fritts, J.E.: Localized Content Based Image Retrieval. IEEE Trans. Pattern Anal. Mach. Intell. 30, 1902–1912 (2008)
7. Maron, O., Lozano-Prez, T.: A Framework for Multiple-Instance Learning. In: 12th International Conference on Advances in Neural Information Processing Systems, pp. 570–576. MIT Press, Cambridge (1998)
8. Ramon, J., De Raedt, L.: Multi Instance Neural Networks. In: ICML 2000 Workshop on Attribute-Value and Relational Learning (2000)
9. Wang, J., Zucker, J.D.: Solving the Multiple-Instance Problem: A Lazy Learning Approach. In: 17th International Conference on Machine Learning, pp. 1119–1126. Morgan Kaufmann, San Francisco (2000)
10. Zhang, Q., Goldman, S.A.: EM-DD: An Improved Multiple-Instance Learning Technique. In: 15th International Conference on Advances in Neural Information Processing Systems, pp. 1073–1080. MIT Press, Cambridge (2001)
11. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support Vector Machines for Multiple-Instance Learning. In: 15th International Conference on Advances in Neural Information Processing Systems, pp. 561–568. MIT Press, Cambridge (2003)
12. Chen, Y., Wang, J.Z.: Image Categorization by Learning and Reasoning with Regions. J. Mach. Learn. Res. 5, 913–939 (2004)
13. Chen, Y., Bi, J., Wang, J.Z.: MILES: Multiple-Instance Learning via Embedded Instance Selection. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1931–1947 (2006)
14. Li, W.J., Yeung, D.Y.: MILD: Multiple-Instance Learning via Disambiguation. IEEE Trans. on Knowl. and Data Eng. 22, 76–89 (2010)
15. Fu, Z., Robles-Kelly, A., Zhou, J.: MILIS: Multiple Instance Learning with Instance Selection. IEEE Trans. Pattern Anal. Mach. Intell. 33, 958–977 (2011)
16. Chang, C.C., Lin, C.J.: LIBSVM: A Library for Support Vector Machines. Software (2012), http://www.csie.ntu.edu.tw/~cjlin/libsvm
17. Erdem, A., Erdem, E.: Multiple-Instance Learning with Instance Selection via Dominant Sets. In: 1st International Workshop on Similarity-Based Pattern Analysis and Recognition, pp. 177–191. Springer, Heidelberg (2011)

Motivating Retail Marketing Efforts under Fairness Concerns in Small-World Networks: A Multi-agent Simulation

Meng Qingfeng[1], Du Jianguo[1], and Li Zhen[2,*]

[1] School of Management, Jiangsu University, Zhenjiang 212013, China
[2] School of Economics and Management, Jiangsu University of Science and Technology, Zhenjiang 212003, China
{mqf1105,jgdu2005,zhenzi2003}@163.com

Abstract. When manufacturer motivates retailer groups to increase sales efforts with a linear transfer payment contract, we assume that retailers concern about fairness and the channel structure which retailer access to information corresponds to the small-world network, and built a multi-agent model to mainly observe the impact of small-world network characteristics on the incentive effects. Experimental results show that the greater probability of replacement objects, the lower manufacturer's profit and products sales. The retailer gets very small amount of other retailers related information will have a huge negative impact on incentive effects, if the number of objects in comparison achieves a certain number, the manufacturers profit and products sales will not be affected to a large extent.

Keywords: Sales Efforts, Linear Transfer Payment Contract, Fairness Preference, Small-world Network, Multi-agent Simulation.

1 Introduction

In most situations, retailer sales efforts are important in influencing demand [1]. Since the agents efforts are more difficult it is to monitor generally, the agents effort level is not a contractible variable. [2] designed a linear transfer payment contract to coordinate the supply chain and achieve a Win-Win outcome. The supply chain structure previously described in the literature is one-to-one, in the real world, incentive to subjects of manufacturer are retailer groups, people are not self-interested, rational agents, however, recent developments in behavioral economics suggest that actors may care about fairness in addition to economic benefits [3,4]. In supply chain, the members have sense of fairness when care about others welfare, some literature assumes that the income information is completely known for all members [5,6]. But in reality, retailers are unlikely to obtained the revenue and cost information of all other retailers in the same

[*] Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 68–75, 2012.

industry. So we assume that the channel structure which agents access to relevant information corresponds to the small-world network structure. Small-world properties are found in many real-world phenomena, such as retail networks.

When manufacturer need to stimulate retailer groups increasing sales efforts with a linear transfer payment contract, and retailers concern about fairness in the distribution, and the channel structure which retailer access to revenue and cost information corresponds to the small-world network structure, how will the properties of small-world network impact the incentive effects? The objective of this paper is to clarify the impact mechanisms, and help manufacturer improve the incentive effect for the practice management, allocate retail networks and take advantage of the communication channels between retailers reasonably, so as to improve its own profit performance. This paper construct an agent-based simulation model to mainly observe the impact of small-world network characteristics on the incentive effect when the agents with inequity aversion, as supply chain systems involving many heterogeneous agents who have autonomous decision-making capacity. The remainder of this paper is structured as follows. Section 2 we illustrate our design for agents attributes and decision-making. Section 3 describes agents interactive in system and operation process of system. Section 4 presents our experiment scenarios and initial parameters setting in simulation. Section 5 analyzes the results of the simulation studies. Section 6 summarizes the insights gained from this study.

2 Multi-agent Design

2.1 Consumer Agent

We consider a model which involved one manufacturer, M retailers and N consumers. Consumers make different purchasing decisions according to threshold utility, u_c^idenotes threshold utility of consumer i, if sales efforts of retailer i exceeds u_c^i, consumer i will make a purchase. Otherwise, consumer i will not make a purchase in this period. In reality, consumer i can not compare all retailers when make purchasing decisions. So we assume that consumers search and compare retailers within a range (a_c^i), consumers i will compare some number of retailers within its search range. Some consumers tried out a product or obtained information about it at one retail store but ended up buying the product at another store. This pattern of consumer behavior gives rise to the well-known free riding phenomenon that occurs in a multi-channel supply chain [7]. So this model assumes that have a certain proportion (p_c) of all consumers who have a purchase motivation or decision that buy product from the retailer who has the best sales efforts, and have $(1-p_c)$ proportion consumers make a purchase from the retailer who be selected at random from their search range (a_c^i). After all consumers make decisions on whether to purchase or not, retailers will achieve their actual product sales accordingly.

2.2 Manufacturer Agent

The manufacturer signed linear transfer payment contract with its all retailers. Before selling season, the manufacturer offers a sales target to retailer, if the final sales quantity is above the target, the manufacturer gives the retailer a rebate; otherwise, the retailer gives a payment to the manufacturer as penalty. The transfer payments $(T(s_r^i))$ between manufacturer and retailer as follows:

$$T(s_r^i) = \chi(s_r^i - s_r^T) \ . \tag{1}$$

We use s_r^i as actual sales of retailer i and s_r^T as sales target which determined by manufacturer. Let χ be the transfer payments coefficient, denote that the size of the transfer payments for each additional unit of sales when the retailer's sales more than (or less than) the sales target. Manufacturer sells products to retailer i through channel i, so the profit (π_m^i) manufacturer gets from channel i as follows:

$$\pi_m^i = (w_m - c_m)s_r^i - T(s_r^i) \ . \tag{2}$$

Let w_m be the manufacturer's wholesale price, c_m be the manufacturer's production cost, π_m be the manufacturer's total income which is the sum of profit from all channels, that is $\pi_m = \sum_{i=1}^{M} \pi_m^i$. The manufacturer's sales target in every period is proportional $(\eta)$to the total sales of all retailers, namely $s_r^T = \frac{\eta \sum_{i=1}^{M} s_r^i}{M}$.

2.3 Retailer Agent

Let p be the retail price of all retailers, as a result of competition. Let c_r^i be the marginal cost per unit of product of retailer i, we use e_r^i as the sales efforts of retailer i, to summarize the retailer's activities in promoting sales and let $g_r^i(e_r^i)$ be the retailer i's cost of exerting a efforts level e_r^i , $g_r^i(e_r^i) = \frac{k_r^i (e_r^i)^2}{2}$, k_r^i are positive constants, denote the relationship between sales efforts and sales cost. There are differences in the marginal cost and promotional cost between retailers, namely c_r^i , k_r^i all difference between retailers, in order to describe inherent ability discrepancies between retailers. Let π_r^i be the actual profit of retailer i as follows:

$$\pi_r^i = (p - w_m - c_r^i)s_r^i - \frac{k_r^i (e_r^i)^2}{2} + T(s_r^i) \ . \tag{3}$$

In real life, retailers can not obtain the related information of all competitors in the same industry, this paper assumes that the channels structure for retailers accessing to information conforms to small-world network structure. Small-world networks, characterized by relatively numerous short-range interconnections along with a few long-range contacts, it is complex networks research results in recent years and widely exist in the real network. A small-world network can be generated by regular networks with q nodes, these nodes are connected in a ring, and each node is symmetrically connected with its k nearest neighbors, so have k edges, and assumes that $q \geq k \geq lnq \geq 1$ in general. Each edge keeps an endpoint unchanged;

rewire the other endpoint to a different node with p_r probability, so we can make networks from completely regular network (p_r=0) to completely random network (p_r=1) transformation by adjusting p_r value. A small-world network with q nodes generated as described above has a total of $q \times k$ links, which is in the order of k [8].

Each retailer can obtain related information of others which keep connected in network. This paper adopts FS inequity aversion model [9] which is used widely to describe retailer i's fairness utility, let f_r^i denote it. In FS model, the fairness utility function considers only the gap between agents' revenue, our model do not focus on the revenue gap, but consider the revenue to cost ratio of agents, and the fairness utility function considers the gap between agent's ratio. So retailer i will compare own revenue to cost ratio $(\frac{\pi_r^i}{c_r^i(T)})$ to other retailers' average revenue to cost ratios $(\frac{\sum_{j\neq i}[\pi_r^j/c_r^j(T)]}{k})$ who have connections in network, and get fairness utility (f_r^i) as follows:

$$\begin{aligned} f_r^i = \frac{\pi_r^i}{c_r^i(T)} - \alpha_r^i &\left[max(\frac{\sum_{j\neq i}^{M} \pi_r^j/c_r^j(T)}{k} - \frac{\pi_r^i}{c_r^i(T)}, 0) \right] \\ -\beta_r^i &\left[max(\frac{\pi_r^i}{c_r^i(T)} - \frac{\sum_{j\neq i}^{M} \pi_r^j/c_r^j(T)}{k}, 0) \right] \end{aligned} \tag{4}$$

In formula (4), $\pi_r^i(\pi_r^j)$ denotes the actual profit of retailer$i(j)$, $c_r^i(T)$ denotes the total cost of retailer i, where α_r^i (β_r^i)measures the retailer i's disutility of revenue to cost ratio less than (more than) its competitors. According to the economic experimental results, past research has shown that "subjects suffer more from inequity that is to their monetary disadvantage than from inequity that is to their monetary advantage" [10], we further assume $\alpha_r^i > \beta_r^i$ and $1 > \beta_r^i \geq 0$, especially, when $\alpha_r^i = \beta_r^i$, means the agent has pure self-interested preferences. Retailer i will adjust the sales efforts according to its fairness utility, the adjustment rules are shown as formula (5).

$$e_r^i(t) = \begin{cases} e_r^i(t-1)\left[1 + co_u(f_r^i(t) - f_r^i(t-1))\right] & \text{if } f_r^i(t) \geq f_r^i(t-1) \\ e_r^i(t-1)\left[1 - co_u(f_r^i(t-1) - f_r^i(t))\right] & \text{if } f_r^i(t) < f_r^i(t-1) \end{cases} \tag{5}$$

$e_r^i(t)$ denotes sales efforts of retailer i in t period, and $f_r^i(t)$ denotes fairness utility of retailer i in t period, co_u denotes adjustment coefficient, for describing the impact of fairness utility on the sales efforts.

3 Process of Operation and Decision-Making

In the first period of experiment, retailers have an initial value of sales efforts, therefore consumers first decide whether to buy and finish their purchasing behavior, so it generated total sales of all retailers and actual sales of each retailer. The manufacturer is responsible for contract parameters, the manufacturer decides the sales target (s_r^T) for next season's sales according to the actual total sales of the products. Retailers can calculate their actual profit according to own actual sales volumes, the manufacturer's wholesale price (w_s) and reward and punishment (χ). Then retailers can calculate their fairness utility by using fairness evaluation function and fairness aversion model, and adjust sales efforts accordingly. When retailers complete making decision regarding sales efforts prepared for the next sales season, consumers make their purchasing decisions again, and the interaction process is the same as described above. After the end of each sales season, the channels which accessing to information reconnection between retailers, they will be determined by the operating rules of small world network.

4 Experiment Scenarios and Initial Parameters Setting

Initialization settings of the basic parameters in experiment are shown in Table 1. We designed two experiment scenarios: First, research on the impact of rewiring probability (p_r) on the incentive effects, we keep the number of nodes (k=10)connected to each retailer constant,p_r increases in steps of 0.1 from 0 to 1 in each experiment, for each p_r, Experiments are scheduled to run during 4000 periods, and statistical analysis of experimental data. Second, research on the impact of the number of nodes (k) connected to each retailer on the incentive effects, we keep rewiring probability (p_r=0.4) constant, set the number of connection nodes (k) in turn to take 2, 10, 20,30,40,50,60,70,80,90 respectively, Experiments are scheduled to run during 4000 periods also for each k value. All the other parameters keep unchanged during two experiments. In order to obtain the necessary experimental data, the program will be repeated fifteen times for average.

Table 1. Initializations of the experimental parameters

Parameters	Scale	Parameters	Scale	Parameters	Scale
M	100	N	10000	χ	2
u_c^i	R[1,100]	a_c^i	3	p_c	0.6
w_m	8	c_m	1	η	0.8
p	20	c_r^i	R[1,3]	e_r^i	R[50,100]
k_r^i	R[0.001,0.01]	a_r^i	R[0.5,1]	β_r^i	R[0,0.5]
k	10	p_r	0.4	co_u	0.5

5 Experimental Results and Analysis

5.1 The Impact of Random Rewiring Probability on Incentive Effects

The manufacturer's profit and products sales under different rewiring probabilities (p_r) are shown in Fig.1. As rewiring probability between retailers in the small world network increasing, the manufacturer's profit (Fig.1(a)) as well as products sales (Fig.1(d)) showed a downward trend. As rewiring probability increases, it means that the probability of retailer's replacement objects in comparison after the end of each sales period increasing, so replace the objects in comparison frequently will have a negative effect on products sales and profit of manufacturer.

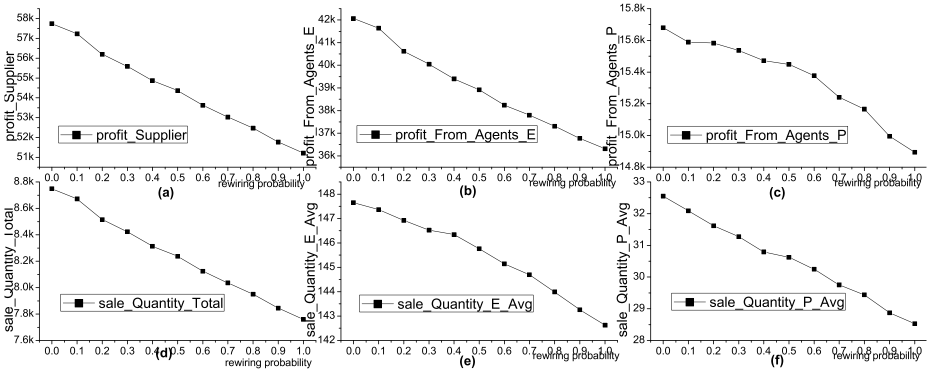


Fig. 1. The manufacturer's profit and sales under different rewiring probabilities

When the transfer payments were completed between agents, retailer groups were divided into two groups which were rewarded (E) and punished (P). Fig.1(b) and 1(c) are shown that the manufacturer obtains the profit derived from rewarded and punished retailers respectively, Fig.1(e) and 1(f) are shown that the average products sales of two groups of retailers above respectively. It can be seen from the figures, when p_r =0, network is completely regular, retailers' average sales efforts are the highest level, it means that the retailers' average fairness utility are maximal. However, with p_r increasing, while the network change from completely regular network toward to completely random network, the retailers' average fairness utility decline gradually, they feel more and more unfair due to replace the objects in comparison frequently. Thus, when the manufacturer motivate retailer groups increasing the sales efforts using linear transfer payment contract, the retailers have unfairness aversion preferences, should limit the channels for retailers accessing to competitors' revenue and cost information within a certain range, can not replace the objects in comparison frequently. Manufacture should try his best to keep channels structure for retailers accessing to information conforms to completely regular network, it means that retailers could know their neighbors' revenue and cost information, and not to know other retailers' related information expect their neighbors.

5.2 The Impact of Number of Connected Nodes on Incentive Effects

Fig.2 shows the manufacturer's profit and products sales under different number of connected nodes (k). When each retailer can get two other retailers' revenue and cost information (k=2), the manufacturer's profit and products sales are very low. Compared to the condition of k=2, manufacturer's profit and sales increase significantly when k=10. Manufacturer's profit has been a slowly rising trend when $k \leq 40$. Experimental results showed that if retailers have unfairness aversion preference and their channels structure for accessing to information conforms to small-world network characteristics, manufacturer need to keep the number of connected nodes maintain the principle of proportionality to have better incentive effects. The retailer gets very small amount of other retailers' related information will have a huge negative impact on his fairness utility, so cause sales efforts reduction further. However, if the number of objects in comparison of retailers achieves a certain number, the manufacturer's profit and products sales will not be affected to a large extent.

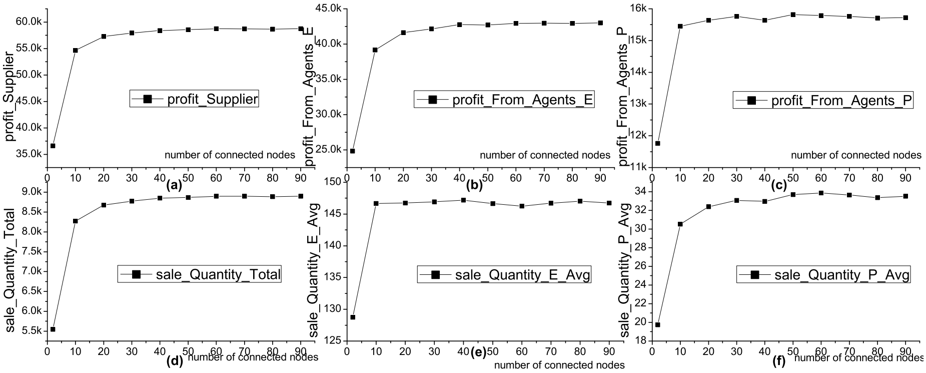


Fig. 2. The manufacturer's profit and sales under different number of connected nodes

Therefore, manufacturer should keep the information transfers between retailers maintain the principle of proportionality during the incentive process, while the motivated objects have unfairness aversion.When the number of objects in comparison reaches a certain level, many indicators in our experiments remain within a certain level. So the manufacturer need to measures to keep the number of objects in comparison is not too small.

6 Conclusions

When retailer sales efforts are important in influencing demand, manufacturer needs to stimulate retailer groups increasing sales efforts with a linear transfer payment contract, this paper assumes that retailers concern about fairness in the distribution and the channel structure which retailer access to revenue and cost information corresponds to the small-world network structure. We construct an agent-based simulation model to mainly observe the impact of small-world network characteristics on the incentive effect when the agents with inequity aversion. This paper designed two experiment scenarios research on the impact of the probability of replacement objects and the number of objects in comparison on incentive effects. Experimental results show that the greater probability of replacement objects, the lower manufacturer's profit and products sales, the manufacturer should not replace the objects in comparison frequently. The retailer gets very small amount of other retailers' related information will have a huge negative impact on incentive effects, if the number of objects in comparison of retailers achieves a certain number, the manufacturer's profit and products sales will not be affected to a large extent.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under grant 71171099, 71073070, 71101067, 71101069, 71001028, 71001049, and by the National Planning Office of Philosophy and Social Science under Grant 11&ZD169, National Social Science Foundation of China under Grant 10CGL025, social science and humanity on Young Fund of the ministry of Education (12YJCZH020).

References

1. He, Y., Zhao, X., Zhao, L.D., He, J.: Coordinating a Supply Chain with Effort and Price Dependent Stochastic Demand. Appl. Math. Model. 33, 2777–2790 (2009)
2. Taylor, T.A.: Supply Chain Coordination under Channel Rebates with Sales Effort Effect. Manag. Sci 48, 992–1007 (2002)
3. Kahneman, D., Knetsch, J.L., Thaler, R.: Fairness, Competition on Profit Seeking: Entitlements in the Market. American Econ. Rev. 76(4), 728–741 (1986)
4. Ruffle, B.J.: More Is Better, But Fair is Fair: Tipping Indictator and Ultimatum Games. Games Econom. Behav. 23, 247–265 (1998)
5. Cui, T.H., Raju, J.S., Zhang, Z.J.: Fairness and Channel Coordination. Manag. Sci. 53(8), 1303–1314 (2007)
6. Loch, C.H., Wu, Y.Z.: Social Preferences and Supply Chain Performance: An Experimental Study. Management Science 54(11), 1835–1849 (2008)
7. Bernstein, F., Song, J.S., Zheng, X.N.: Free Riding in a Multi-Channel Supply Chain. Naval Research Logist 56, 745–765 (2009)
8. Man, S.S., Hong, D.W., Michael, A.P., Joseph, V.M.: A Computational Model for Signaling Pathways in Bounded Small-World Networks Corresponding to Brain Size. Neurocomput. 74, 3793–3799 (2011)
9. Fehr, E., Schmidt, K.M.: A Theory of Fairness, Competition and Cooperation. Quart. J. Econom. 114, 817–868 (1999)
10. Fehr, E., Gachter, S.: Cooperation and Punishments in Public Goods Experiments. American Econom. Rev. 90, 980–994 (2000)

Application of Variational Granularity Language Sets in Interactive Genetic Algorithms

Dunwei Gong, Jian Chen, Xiaoyan Sun, and Yong Zhang

School of Information and Electrical Engineering,
China University of Mining and Technology, Xuzhou, China
dwgong@vip.163.com, chenjian121206@163.com, {xysun78,yongzh401}@126.com

Abstract. An interactive genetic algorithm with evaluating individuals using variational granularity was presented in this study to effectively alleviate user fatigue. In this algorithm, multiple language sets with different evaluation granularities are provided. The diversity of a population described with the entropy of its gene meaning units is utilized to first choose parts of appropriate language sets to participate in evaluating the population. A specific language set for evaluating an individual is further selected from these sets according to the distance between the individual and the current preferred one. The proposed algorithm was applied to a curtain evolutionary design system and compared with previous typical ones. The empirical results demonstrate the strengths of the proposed algorithm in both alleviating user fatigue and improving the efficiency in search.

Keywords: Genetic Algorithm, Interaction, User Fatigue, Granularity, Entropy.

1 Introduction

Interactive genetic algorithms (IGAs), proposed in the mid 1980s, are effective methods suitable for handling optimization problems with qualitative indices, however, the user may fail to perform the evolutionary process to reach a satisfactory solution due to fatigue. So developing appropriate methods to effectively alleviate user fatigue is of great significance for IGAs.

Some studies have been motivated to alleviate user fatigue of IGAs, and they can roughly be classified into three categories. The first is to construct surrogate models to approximate the user's evaluations and employ them in the subsequent evolutions to estimate the fitness of all or a part of individuals instead of the user[1,2]. The second is to reduce the number of human-computer interactions by accelerating the process of searching for satisfactory solutions with improved genetic operators[3]. And the last is to design friendly interfaces or adopt more natural evaluation styles to directly reduce the evaluation burden of the user[4,5].

In view of the user's fuzzy cognition to individuals, we utilized a fuzzy number to express an individual's fitness[5]. However, the distinguishable degree of the evaluations for subtle individuals' differences, is same. Intuitively, a language set

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 76–83, 2012.

with a smaller number of linguistic values has a rougher or larger granularity and vice versa. In fact, in IGAs, adopting language sets with variational granularity to evaluate a population with different distributions and individuals in diverse areas is necessary, which can well balance the burden and the accuracy of the user's evaluations.

To this end, an IGA with variational granularity of individuals' evaluations (IGA-VGIE) was presented. Multiple language sets with different evaluation granularities are used to evaluate individuals; several language sets suitable for evaluating the current population are first selected in terms of its diversity, and then a specific set for evaluating an individual in the population is determined according to its distance to the current preferred one. The advantage of this algorithm is that when the population has good diversity, language sets with a rough granularity will be preferably employed to evaluate it in order to alleviate the user's evaluation burden. In contrast, when the diversity of the population is bad and its individuals are similar to the current preferred one, language sets with a fine granularity will be utilized so that a small difference among these individuals can well be distinguished and accurate evaluations are easily obtained. Consequently, the evaluation burden and accuracy are well balanced.

The main contributions of this work are: 1) proposing a method of quantitatively describing the diversity of a population with the entropy of gene meaning units, and hence getting an interval to which the diversity belongs; 2) presenting an approach to quantitatively describing the discerning capability of a language set based on the number of linguistic values belonging to it, and further obtaining the membership function that describes the evaluation applicability of a language set; 3) giving a strategy of determining the evaluation granularity for an individual based on the diversity and its distance to the current preferred one.

2 Proposed Algorithm

The following optimization problem was considered:

$$\begin{array}{ll} \max & f(x) \\ \text{s.}t. & x \in S \subseteq R^n. \end{array} \tag{1}$$

where $f(x)$ is a performance index to be optimized, x is an n-dimensional variable belonging to domain S. In this work, an IGA is used to solve the above problem, the corresponding individual and the search space are also denoted as x and S, respectively.

2.1 Diversity of Population

In our study, the entropy of each gene meaning units was defined to depict the diversity of a population[6]. Without loss of generality, decision variables encoded with binary bits were considered here. Denote a population, with the population size of N, in the t-th generation as $x(t)$, the i-th gene meaning unit as $U_i(t)$, the allelic gene meaning unit of $U_i(t)$ for the j-th individual as $U_i^j(t)$ and the number of individuals with the same value of $U_i^j(t)$ as $\alpha_i^j(t)$. Then $\alpha_i^j(t)$ can be calculated with the following formula:

$$\alpha_i^j(t) = \sum_{q=1}^{N} n(U_i^j(t), U_i^q(t)), \tag{2}$$

where

$$n(U_i^j(t), U_i^q(t)) = \begin{cases} 1 & U_i^j(t) = U_i^q(t), \\ 0 & U_i^j(t) \neq U_i^q(t). \end{cases} \tag{3}$$

The entropy of $U_i(t)$, designated as $H(U_i(t))$, can be expressed as follows:

$$H(U_i(t)) = -\frac{1}{N}\sum_{j=1}^{N} \log_2 \frac{\alpha_i^j(t)}{N}\,. \tag{4}$$

Evidently, from equations (2) and (4), we can conclude that $H(U_i(t)) \in [0, \log_2 N]$. Normalize $H(U_i(t))$ and denote it as $H'(U_i(t))$, which can be defined as follows:

$$H'(U_i(t)) = \frac{H(U_i(t))}{\log_2 N} \tag{5}$$

An interval was adopted to reflect the diversity in this work. Denote the total number of gene meaning units in an individual as n_U and the diversity of $x(t)$ as $D(x(t))$, which can be reflected with an interval determined by $\min\limits_{i\in\{1,2,\cdots,n_U\}} H'(U_i(t))$ and $\max\limits_{i\in\{1,2,\cdots,n_U\}} H'(U_i(t))$ as:

$$D(x(t)) \in [\min_{i\in\{1,2,\cdots,n_U\}} H'(U_i(t)), \max_{i\in\{1,2,\cdots,n_U\}} H'(U_i(t))]. \tag{6}$$

2.2 Selection of Language Sets

Discrimination of Language Set. The discrimination of a language set on evaluations was first defined with entropy. Obviously, the more the linguistic values included in a language set, the stronger the discrimination it has on evaluating individuals.

Denote the number of language sets as n_F, the i-th language set as $S_i(\ i = 1, 2, \cdots, n_F)$ with linguistic values of $s_i^1, s_i^2, \cdots, s_i^{|S_i|}$. Assume that the sizes of these sets satisfy $|S_1| \leq |S_2| \leq \cdots \leq |S_{n_F}|$. If a language set is treated as an information source, the linguistic values contained in the language set can be regarded as the messages of this information source. For S_i, the selection frequency of any linguistic value is $\frac{1}{|S_i|}$, consequently, the entropy of this language set can be calculated as follows:

$$H(S_i) = -\sum_{k=1}^{|S_i|} p(s_i^k) \log_2 p(s_i^k) = \log_2 |S_i|\,. \tag{7}$$

It can easily be observed from equation (7) that S_{n_F} has the strongest discrimination. If the discrimination of S_{n_F} was denoted as one, the discrimination of fuzzy language set S_i, denoted as $I(S_i)$, can be described as follows:

$$I(S_i) = \frac{H(S_i)}{H(S_P)}. \tag{8}$$

Language Sets for Population Evaluation. To get the required language sets, the adaptability of a language set in evaluating a population was first presented. Evidently, it is a fuzzy concept and can thus be expressed with a membership function. Denote the adaptability of language set S_i in evaluating $x(t)$ as $F_{S_i}(x(t))$ which has a close relationship with $D(x(t))$, as analyzed before, i.e., $F_{S_i}(x(t))$ can be formulated as a function of $D(x(t))$. So it is nature to take $F_{S_i}(x(t))$ as a membership function defined in the range of $[0, 1]$.

For convenience to illustrate, here four language sets were used, i.e., $n_F = 4$ and $|S_1| = 3$,$|S_2| = 5$,$|S_3| = 7$, $|S_4| = 9$. Because of $I(S_1) = 0.5$, $I(S_2) = 0.732$, $I(S_3) = 0.886$ and $I(S_4) = 1$, S_4 has the finest granularity among the four sets, and is only required to evaluate a population when the diversity of the population is very low. Accordingly, its adaptability reaches the maximum in the anaphase. Since S_1 only has a half discrimination of S_4, the adaptability of S_1 will be high for a diverse population, especially for the population in the initial generation. As for S_2 and S_3, since their discriminations lie in between S_1 and S_4, their adaptabilities can be the maximal when the diversity of the population reaches at a certain level and decreases along with the decrease (or increase) of the diversity of the population at the two sides of that level. Motivated by these analyses, $F_{S_i}(x(t))(i = 1, 2, 3, 4)$ are illustrated in Figure 1.

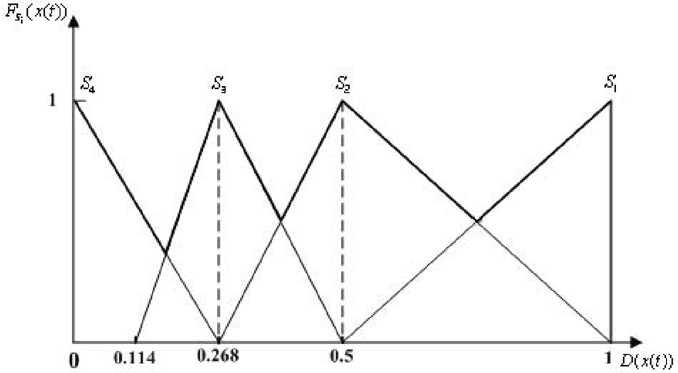


Fig. 1. Adaptability of language sets

To choose appropriate language sets to evaluate the current population, the adaptabilities of each language set to both limits of $[\min_{i\in\{1,2,\cdots,n_U\}} H'(U_i(t))$,

$\max\limits_{i\in\{1,2,\cdots,n_U\}} H'(U_i(t))]$ to which the diversity of the population belongs are calculated. Language sets with the maximal adaptability to both limits are first selected; further, other sets whose granularities fall in between those of the two selected sets are also picked up.

Language Set for Individual Evaluation. Having selected language sets for evaluating a population, the method of choosing a specific set to evaluate an individual in the population was presented in this subsection. All individuals in the current population are first classified into several clusters with the nearest neighbor-based method, and the number of clusters is equal to that of language sets used to evaluate the population. Then, according to the distance between the center of a cluster and the preferred individual, a specific language set for evaluating all individuals in this cluster can be collected.

For the current population $x(t)$, denote the number of fuzzy language sets selected for evaluating it as n_F, namely, $S_1, S_2, \cdots, S_{n_F}$. The current $x(t)$ is first classified into n_F clusters. To this end, denote the most preferred individual in $x(t)$ as $x^B(t)$, and the distance between the j-th individual in $x(t)$ and $x^B(t-1)$ as $d_j(t)$, then the center individual of the i-th cluster, designated as $x_i^{\mathrm{C}}(t)$, can be expressed as follows.

$$x_i^{\mathrm{C}}(t) = \arg\left\lfloor \max_{j=1,2,\ldots,N} d_j(t) - \tfrac{i-1}{n_F-1}\cdot\left(\max_{j=1,2,\ldots,N} d_j(t) - \min_{j=1,2,\ldots,N} d_j(t)\right)\right\rfloor \cdot \quad (9)$$
$$(\ i = 1, 2, \cdots, n_F)$$

The language set with the finest granularity is picked up from language sets for evaluating $x(t)$ to evaluate the cluster closest to $x^B(t-1)$, i.e., the cluster with the largest i. The granularity of the language set for evaluating individuals increases along with the decrease of i, consequently, the language set used to evaluate individuals in the 1-st cluster has the maximal granularity among all sets for evaluating $x(t)$.

3 Application in Evolutionary Curtain Design System

An evolutionary curtain design system was developed based on the framework of our algorithm. In addition, some other algorithms, including IGA-VGIE, TIGA(i.e., an individual's fitness is expressed by an exact value,), IGA-SLE3 (i.e., the IGA-SLE$|S|$ whose $|S| = 3$) and IGA-SLE9 (i.e., the IGA-SLE$|S|$ whose $|S| = 9$) were compared with the proposed one according to their applications in the system to fully demonstrate the advantages of our algorithm in both alleviating user fatigue and improving the search ability.

3.1 Backgrounds and Encoding

The evaluation of a user to a curtain depends mainly on its appearance, characterized with the corresponding pattern and color. Obviously, it is hard to find a uniform and explicit objective to evaluate a design, suggesting that traditional GAs cannot solve this problem, whereas IGAs can.

In our evolutionary curtain design system, the binary encoding scheme was adopted for all comparative algorithms to represent the genotype of an individual. Since a curtain is made up of three independent parts, each part is treated as a gene meaning unit and can be expressed with a six-bit binary string, the total length of the genotype of a curtain is 18. Specifically, the first six-bit represents the style of top, the 6th to 11th bits expresses the drapery and the last six-bit expresses the crossover leno. With the aforementioned encoding, there are total $2^{18} = 262144$ curtains to be explored by an IGA in this system.

3.2 Parameter Settings

The population size of all algorithms was set as 100, but the user only evaluated eight individuals in each generation, the other ones were automatically estimated by the system. For IGA-VGIE, according to the clustering results, l_i individuals were selected from cluster $\{x_i^{\mathrm{C}}(t)\}$ to be evaluated by user, satisfying $\frac{l_1}{|\{x_1^C(t)\}|} = \frac{l_2}{|\{x_2^C(t)\}|} = \cdots = \frac{l_{n_F}}{|\{x_{n_F}^C(t)\}|}$ and $\sum_{i=1}^{n_F} l_i = 8$. The fitness of the rest individuals were estimated based on the evaluated ones. According to the transformation form presented in [7] for comparing linguistic values with different granularities, the estimate of the fitness contained two parts, i.e., a linguistic value s and its deviation α. For the other three comparative algorithms, individuals with the same value of $d_i(t)$ are classified into the same cluster, and l_i individuals are randomly selected from cluster $\{x_i^{\mathrm{C}}(t)\}$ to be evaluated by the user.

The same genetic operators and parameters were adopted in all algorithms, specifically, tournament selection with size of 2, one-point crossover and one-point mutation. The probabilities of crossover and mutation, p_c and p_m, were 0.6 and 0.05, respectively, and the maximal number of generations was 20, i.e., $T_{\max} = 20$. The algorithm would automatically be stopped after the population had evolved for $T_{\max}$ generations or manually be stopped if the user had got the most satisfactory curtain. Four language sets with different numbers of linguistic values were used in our algorithm, i.e., $|S_1| = 3$,$|S_2| = 5$,$|S_3| = 7$, $|S_4| = 9$.

3.3 Results and Analysis

To compare the performances of different algorithms, the evolutionary curtain design system was independently run 20 times for each algorithm, and the number of individuals searched by the system, the number of generations and the time consumed by the user were recorded. Besides their averages and variances (as listed in Table 1), t-test results were also provided to determine whether there are significant differences on these indices between the proposed algorithm and the comparative one. Suppose that the null hypothesis for the number of individuals searched by the system are $H_0 : \mu_1 > \mu_2$ and $H_1 : \mu_1 \leq \mu_2$, and for the number of generations and the time consumed by the user are $H_0 : \mu_1 \leq \mu_2$

Table 1. Number of individuals searched by system, number of generations and time consumed by user

	number of individuals searched by system		number of generations		time consumed by user(s)	
	average	variance	average	variance	average	variance
IGA-VGIE	89.78	20.61	13.20	2.06	204.30	536.01
TIGA	86.35	16.33	16.15	1.53	525.35	534.93
IGA-SLE3	76.95	21.75	16.55	2.05	342.95	1203.45
IGA-SLE9	84.45	18.95	15.50	1.45	381.15	805.23

Table 2. Results of t-test

	number of individuals searched by system	number of generations	time consumed by user(s)
TIGA VS. IGA-VGIE	2.46	6.79	42.18
IGA-SLE3 VS. IGA-VGIE	8.59	7.20	14.49
IGA-SLE9 VS. IGA-VGIE	3.69	5.35	21.05

and $H_1 : \mu_1 > \mu_2$, where μ_1 is the average of the comparative algorithm and μ_2 that of the proposed algorithm. The significance level was set as 0.05, so the rejection region of the null hypothesis is $t > t_{1-\alpha}(m+n-2) = 1.684$. The results of the t-tests were given in Table 2.

As can be observed from data in Table 1,

(1) For the number of individuals searched by the system, the proposed algorithm obtains the most, 89.78, followed by TIGA with 86.35, IGA-SLE9 with 84.45 and IGA-SLE3 with 76.95. The t-test results listed in the first column of Table 2 demonstrate that the number of individuals searched by the system using IGA-VGIE is significantly larger than those using the other three algorithms.

(2) For the number of generations, the proposed algorithm requires the least, 13.20, whereas IGA-SLE3 has the most, 16.55, and IGA-SLE9 needs slightly smaller than TIGA. The t-test results listed in the second column of Table 2 show that there are significant differences between the proposed algorithm and the comparative ones, suggesting that our algorithm outperforms the other three in the number of generations.

(3) The time consumed by the user employing the proposed algorithm is 204.30”, the least among all algorithms, TIGA is the largest, 525.35”, and IGA-SLE9 is slightly larger than IGA-SLE3. The time spent by the user using the proposed algorithm is significantly smaller than those spent by the other three, which can easily be obtained based on data in the third column of Table 2.

To sum up, the proposed algorithm finds the most individuals with the smallest generations and the shortest time consumed by the user in evaluations, suggesting that the proposed algorithm is remarkably improved in the search ability and alleviating user fatigue.

4 Conclusions

A novel IGA adopting language sets with variational granularity to evaluate individuals was investigated in this study. According to the diversity of the population and the distance between the unevaluated individual and the preferred one of the user to pick up a specific language set for evaluating an individual, the proposed algorithm remarkably benefits to well balance the burden and the accuracy of evaluations.

While the triangle membership function was designed to describe the adaptability of a language set, other types of functions, e.g., bell or trapezoid, can potentially be adopted, which will produce different results in selecting language sets to evaluate a population. Furthermore, if other ways are used to divide the population, an individual's fitness may also change. These issues will further be investigated in the future.

Acknowledgments. This work was supported by National Nature Science Foundation of China with grant No.61105063 and No.60775044, Natural Science Foundation of Jiangsu Province with grant No.BK2010186 and Post Doctor Foundation of Jiangsu Province with grant No.1001019C.

References

1. Ong, Y.S., Nair, P.B., Lum, K.Y.: Max-min surrogate-assisted evolutionary algorithm for robust design. IEEE Trans. Evol. Comput. 10, 392–404 (2006)
2. Zhou, Z.Z., Ong, Y.S., Nair, P.B.: Combining global and local surrogate models to accelerate evolutionary optimization. IEEE Trans. Sys. Man. Cybernetics-part C: Applications and Reviews 37, 66–76 (2007)
3. Sun, X.Y., Chen, J.: Grid-based knowledge-guided interactive genetic algorithm and its application to curtain design. In: Proceedings of Nature and Biologically Inspired Computing, pp. 395–400. IEEE Press, Kitakyusu (2010)
4. Secretan, J.: Picbreeder: evolving picture collaboratively online. In: Proceedings of Computer Human Interaction Conference, pp. 1759–1768. ACM, New York (2008)
5. Gong, D.W., Yuan, J., Sun, X.Y.: Interactive genetic algorithms with fuzzy individual fitness. Computers. Human. Behavior. 27, 1482–1492 (2011)
6. Gong, D.W., Hao, G.S., Zhou, Y.: Interactive Genetic Algorithms Theory and Application. National Defense Industry Press, Beijing (2007)
7. Herrera, F., Martinez, L.: A model based on linguistic 2-tuples for dealing with multigranular hierarchical linguistic contexts in multi-expert decision-making. IEEE Trans. Sys. Man. Cybernetics-Part B: Cybernetics 31, 227–234 (2001)

Linked PARAFAC/CP Tensor Decomposition and Its Fast Implementation for Multi-block Tensor Analysis

Tatsuya Yokota[1,2], Andrzej Cichocki[2], and Yukihiko Yamashita[1]

[1] Tokyo Institute of Technology, Tokyo, Japan
[2] RIKEN Brain Science Institute, Saitama, Japan
yokota@yy.ide.titech.ac.jp

Abstract. In this paper we propose a new flexible group tensor analysis model called the linked CP tensor decomposition (LCPTD). The LCPTD method can decompose given multiple tensors into common factor matrices, individual factor matrices, and core tensors, simultaneously. We applied the Hierarchical Alternating Least Squares (HALS) algorithm to the LCPTD model; besides we impose additional constraints to obtain sparse and nonnegative factors. Furthermore, we conducted some experiments of this model to demonstrate its advantages over existing models.

Keywords: Tensor decompositions of multi-block data, PARAFAC/CP model, Group Analysis, Hierarchical Alternating Least Squares (HALS).

1 Introduction

The group (multi-block) tensor decomposition is a very important technique in neuroscience, image analysis, and some multi-modal data processing [3,6,11,7]. The group analysis seeks to identify some factors that are common in two or more blocks in a group [3]. The simultaneous tensor decomposition (STD) is known as one of the methods to extract common factor matrices from a group of subjects. The STD model can be applied into tensor based principal component analysis (PCA) and feature extraction for EEG classification [12].

In this paper, we consider a more flexible decomposition model called the linked tensor decomposition (LTD). The LTD method extracts not only their common factor matrices but also their individual (statistically independent) factor matrices at the same time. The LTD model can be characterized as a generalized model of the STD. In fact, it is an intermediate model between the STD model and the individual tensor decomposition model (i.e. standard tensor decomposition of individual blocks).

In order to implement the LTD model, we applied the Hierarchical Alternating Least Squares (HALS) algorithm with the CP (Canonical Polyadic) constraint and two options of sparsity and non-negativity constraints. We call this method the "Linked CP Tensor Decomposition" (LCPTD). Although the CP model is generally unique we will impose some constraints to obtain more meaningful components.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 84–91, 2012.

The rest of this paper is organized as follows. In Section 2, the existing models of tensor analysis are briefly explored. In Section 3, we introduce a novel linked tensor decomposition and its algorithm. In Section 4, we demonstrate experiments using our new method and present the results of these experiments. Finally, we give our conclusions in Section 5.

2 Tensor Decompositions

2.1 Single Tensor Decomposition Based on CP Model

The Canonical Polyadic (CP) model which is also called PARAFAC [8] or CANDECOMP [2] has been well used in positron emission tomography (PET), spectroscopy, chemometrics and environmental science [6,1]. The CP model can be expressed as

$$\underline{\boldsymbol{Z}} \approx \widehat{\underline{\boldsymbol{Z}}} := [\![\underline{\boldsymbol{G}}; \boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ \ldots, \boldsymbol{U}^{(N)}]\!] = \sum_{j=1}^{J} g_j \boldsymbol{u}_j^{(1)} \circ \boldsymbol{u}_j^{(2)} \circ \cdots \circ \boldsymbol{u}_j^{(N)}, \tag{1}$$

where $\underline{\boldsymbol{Z}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is an N-order tensor and model, $\boldsymbol{U}^{(n)} = [\boldsymbol{u}_1^{(n)}, \ldots, \boldsymbol{u}_J^{(n)}]$ $\in \mathbb{R}^{I_n \times J}$ is an n-mode factor matrix with components $\boldsymbol{u}_j^{(n)}$, $\underline{\boldsymbol{G}} = \underline{\boldsymbol{\Lambda}} \in \mathbb{R}^{J \times \cdots \times J}$ is a diagonal core tensor with entries g_j on the main diagonal. The goal of CP decomposition is to estimate factor matrices by minimizing a Frobenius norm of residual tensor $\underline{\boldsymbol{E}} := \underline{\boldsymbol{Z}} - \widehat{\underline{\boldsymbol{Z}}}$. The criterion is given by:

$$\text{minimize} \quad ||\underline{\boldsymbol{E}}||_F^2 = \left|\left| \underline{\boldsymbol{Z}} - \Sigma_{j=1}^{J} g_j \boldsymbol{u}_j^{(1)} \circ \cdots \circ \boldsymbol{u}_j^{(N)} \right|\right|_F^2, \text{ s.t. } ||\boldsymbol{u}_j^{(n)}|| = 1, \tag{2}$$

for $n = 1, \ldots, N$ and $j = 1, \ldots, J$.

When we treat a real-world data, sparsity and non-negativity of factor matrices may play a key role to extract meaningful components. There are many methods for feature extraction and blind source separation using sparsity and non-negativity constraints such as sparse principal component analysis [9] and nonnegative matrix factorization [10,6]. A sparsity constraints are given by $||\boldsymbol{u}_j^{(n)}||_1 < v$, where $|| \cdot ||_1$ is l_1-norm, and v is a threshold parameter. When it is added into (2), then the criterion provides sparse factor matrices. Next the non-negativity constraint is given by $u_{ji}^{(n)} \geq 0, \ g_j \geq 0 \ \forall j, i, n$. In the same way, when the constraints is added into (2), then the criterion realizes nonnegative tensor factorization (NTF).

2.2 Simultaneous Tensor Decomposition

In this section, we introduce the simultaneous CP tensor decomposition (SCPTD). This is very important to explain the proposed method; since the STD is closely related to our new LCPTD. We discuss multiple tensor decompositions from here; besides we assume that there are S tensors of the same dimensions and we obtain S decompositions. We can consider S as the number of blocks (e.g. each block data represents one subject).

One of the objective of group tensor analysis is to decompose individual tensors one by one based on the CP model which is principally unique. We describe this model as the individual CP tensor decomposition (ICPTD). However, in such case the factor matrices are not directly linked.

On the other hand, it is meaningful to extract some common factors for each block which link block by some common factors. The formulation of the SCPTD is given by $\underline{\boldsymbol{Z}}^{(s)} \approx \widehat{\underline{\boldsymbol{Z}}}^{(s)} := [\![\underline{\boldsymbol{G}}^{(s)}; \boldsymbol{U}^{(1)}, \ldots, \boldsymbol{U}^{(N)}]\!] = \sum_{j=1}^{J} g_j^{(s)} \boldsymbol{u}_j^{(1)} \circ \cdots \circ \boldsymbol{u}_j^{(N)}$. The key-point here is that the basis components ($\boldsymbol{u}_j^{(n)}$ of $\boldsymbol{U}^{(n)}$) are the same for all blocks. Only the core tensors $\underline{\boldsymbol{G}}^{(s)}$ are different for individual blocks which represent features [12].

3 Linked CP Tensor Decomposition

In this section, we propose a new model of simultaneous decomposition called the "Linked CP tensor decomposition"(LCPTD) as

$$\underline{\boldsymbol{Z}}^{(s)} \approx \widehat{\underline{\boldsymbol{Z}}}^{(s)} = [\![\underline{\boldsymbol{G}}^{(s)}; \boldsymbol{U}^{(1,s)}, \ldots, \boldsymbol{U}^{(N,s)}]\!] = \sum_{j=1}^{J} g_j^{(s)} \boldsymbol{u}_j^{(1,s)} \circ \ldots \circ \boldsymbol{u}_j^{(N,s)}, \tag{3}$$

where each factor matrix $\boldsymbol{U}^{(n,s)} = [\boldsymbol{U}_C^{(n)}, \boldsymbol{U}_I^{(n,s)}] \in \mathbb{R}^{I_n \times J}$ is composed of two set of bases: $\boldsymbol{U}_C^{(n)} \in \mathbb{R}^{I_n \times L_n}$ (with $0 \leq L_n \leq J$), which is a common factor matrix for all blocks and corresponds to the same or maximally correlated components and $\boldsymbol{U}_I^{(n,s)} \in \mathbb{R}^{I_n \times J - L_n}$, which corresponds to different individual characteristics.

The LCPTD can be considered as a generalized model of simultaneous decomposition. When we put $L_n = J$, its decomposition is equivalent to the simultaneous common factor decomposition [12]. On the other hand, when $L_n = 0$, its decomposition of each subject is equivalent to the standard tensor decomposition. Then the LTD is an intermediate decomposition between simultaneous and normal tensor decomposition.

3.1 LCPTD-HALS Algorithm

In this section, we introduce a new HALS algorithm for LCPTD. Optimization criterion for LCPTD is given by

$$\text{minimize} \quad \sum_{s=1}^{S} \left|\left| \underline{\boldsymbol{Z}}^{(s)} - \sum_{j=1}^{J} g_j^{(s)} \boldsymbol{u}_j^{(1,s)} \circ \ldots \circ \boldsymbol{u}_j^{(N,s)} \right|\right|_F^2, \tag{4}$$

$$\text{subject to} \quad \boldsymbol{u}_j^{(n,1)} = \cdots = \boldsymbol{u}_j^{(n,S)} \text{ for } j \leq L_n, \; ||\boldsymbol{u}_j^{(n,s)}|| = 1, \tag{5}$$

for all n, s, and j. Furthermore, we add $||\boldsymbol{u}_j^{(n,s)}||_1 < v$ or $u_{ji}^{(n,s)} \geq 0$, $g_j^{(s)} \geq 0 \;\; \forall i, j, n, s$ into (5) if we want to get sparse or nonnegative components.

The Hierarchical ALS (HALS) algorithm was first proposed for the Nonnegative Matrix Factorization and Nonnegative Tensor Factorization (NTF) in

[5]. The HALS algorithm were applied to the CP model and it achieved good performances in [4]. In this algorithm, we consider J local-problems and solve them sequentially and iteratively instead of solving (4) and (5), directly. Let $\underline{\boldsymbol{Y}}_j^{(s)} := \underline{\boldsymbol{Z}}^{(s)} - \sum_{i \neq j} g_i^{(s)} \boldsymbol{u}_i^{(1,s)} \circ \ldots \circ \boldsymbol{u}_i^{(N,s)}$, the j-th local problem is given by

$$\begin{aligned} &\text{minimize} \quad \sum_{s=1}^{S} ||\underline{\boldsymbol{Y}}_j^{(s)} - g_j^{(s)} \boldsymbol{u}_j^{(1,s)} \circ \ldots \circ \boldsymbol{u}_j^{(N,s)}||_F^2, && (6) \\ &\text{subject to} \quad \boldsymbol{u}_j^{(n,1)} = \cdots = \boldsymbol{u}_j^{(n,S)} \text{ if } j \leq L_n, \ ||\boldsymbol{u}_j^{(n,s)}|| = 1, && (7) \end{aligned}$$

for all n and s. The LTD-HALS algorithm can be summarized as Algorithm 1; note it does not require matrix inversion and is solved by only simple calculation.

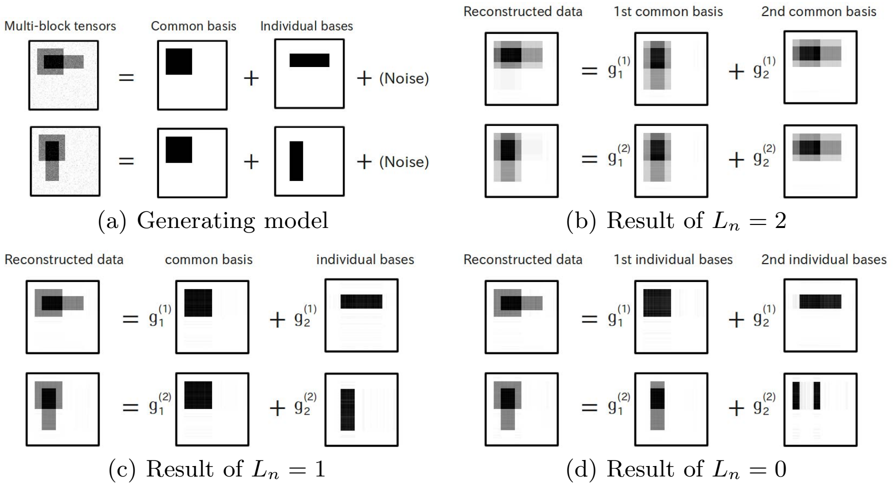
Algorithm 1. LTD-HALS algorithm

```
Input: $\{\underline{\boldsymbol{Z}}^{(s)}\}_{s=1}^{S}$, $J$, and $\{L_n\}_{n=1}^{N}$
Initialize: $\{\boldsymbol{g}^{(s)},\ \{\boldsymbol{U}^{(n,s)}\}_{n=1}^{N}\}_{s=1}^{S}$.
$\underline{\boldsymbol{E}}^{(s)} = \underline{\boldsymbol{Z}}^{(s)} - \Sigma_{j=1}^{J} g_j^{(s)} \boldsymbol{u}_j^{(1,s)} \circ \cdots \circ \boldsymbol{u}_j^{(N,s)}$ for all $s$;
repeat
  for $j = 1, \ldots, J$ do
    $\underline{\boldsymbol{Y}}_j^{(s)} = \underline{\boldsymbol{E}}^{(s)} + g_j^{(s)} \boldsymbol{u}_j^{(1,s)} \circ \cdots \circ \boldsymbol{u}_j^{(N,s)}$ for all $s$;
    for $n = 1, \ldots, N$ do
      Updating $\boldsymbol{u}_j^{(n,s)}$:

        $\boldsymbol{u}_j^{(n,s)} \leftarrow g_j^{(s)} \underline{\boldsymbol{Y}}_j^{(s)} \times_1 \boldsymbol{u}_j^{(1,s)} \cdots \times_{n-1} \boldsymbol{u}_j^{(n-1,s)}$
              $\times_{n+1} \boldsymbol{u}_j^{(n+1,s)} \cdots \times_N \boldsymbol{u}_j^{(N,s)}$ for all $s$;        (8)

      if $j \leq L_n$, $\boldsymbol{t} \leftarrow \sum_{s=1}^{S} \boldsymbol{u}_j^{(n,s)}$;   $\boldsymbol{u}_j^{(n,s)} \leftarrow \boldsymbol{t}$ for all $s$; end if
      Normalize $\boldsymbol{u}_j^{(n,s)} \leftarrow \boldsymbol{u}_j^{(n,s)} / ||\boldsymbol{u}_j^{(n,s)}||$ for all $s$;
    end for
    Update $g_j^{(s)}$:

        $g_j^{(s)} \leftarrow \underline{\boldsymbol{Y}}_j^{(s)} \times_1 \boldsymbol{u}_j^{(1,s)} \cdots \times_N \boldsymbol{u}_j^{(N,s)}$ for all $s$;        (9)

    $\underline{\boldsymbol{E}}^{(s)} = \underline{\boldsymbol{Y}}_j^{(s)} - g_j^{(s)} \boldsymbol{u}_j^{(1,s)} \circ \cdots \circ \boldsymbol{u}_j^{(N,s)}$ for all $s$;
  end for
until $\sum_{s=1}^{S} ||\underline{\boldsymbol{E}}^{(s)}||_F^2$ converge
Output: $\{\boldsymbol{g}^{(s)},\ \{\boldsymbol{U}^{(n,s)}\}_{n=1}^{N}\}_{s=1}^{S}$
```

If we want to obtain sparse components, we implement the following updates after (8):

$$\boldsymbol{u}_j^{(n,s)} \leftarrow \text{sign}(\boldsymbol{u}_j^{(n,s)}) \circledast [\text{abs}(\boldsymbol{u}_j^{(n,s)}) - \xi_n \mathbf{1}]_+ \text{ for all } s; \qquad (10)$$

where ξ_n is a positive parameter deciding on their sparsity.



(a) Generating model

(b) Result of $L_n = 2$

(c) Result of $L_n = 1$

(d) Result of $L_n = 0$

Fig. 1. Linked Multi-block Tensor Factorization

If we want to obtain nonnegative components, we implement the following updates after (8) and (9):

$$\boldsymbol{u}_j^{(n,s)} \leftarrow [\boldsymbol{u}_j^{(n,s)}]_+ \text{ for all } s, \tag{11}$$

$$g_j^{(s)} \leftarrow [g_j^{(s)}]_+ \text{ for all } s. \tag{12}$$

4 Experiments

4.1 Toy Problem for Linked Multi-block Tensor Factorization

In this part, we applied the LCPTD to a toy problem (benchmark) for linked multi-block tensor factorization. We generated two block data tensors consisting of a one common basis factor and two individual basis factors with noise (see Fig. 1(a)). And we decompose them by our LCPTD model with nonnegative constraints for various number of common bases $L_n \in \{2, 1, 0\}$ for $n = 1, 2$. Fig. 1(b,c,d) depict the results of this experiment. It is obvious that the result of $L_n = 2$ couldn't represent the original data tensors since the degree of freedom of model is not sufficient. On the other hand, the result of $L_n = 0$ could represent the original data tensors, but each basis is not matched, completely. The result of $L_n = 1$ could represent not only the original data tensors, but also each basis; besides, the additive noise were reduced.

We can see that the LCPTD model can be very useful assuming that some components are common in generating model. The blind source separation can separate into two original sources from two observed signals. It is very interesting that the LCPTD can achieve separation of three bases (i.e., a common and two individual

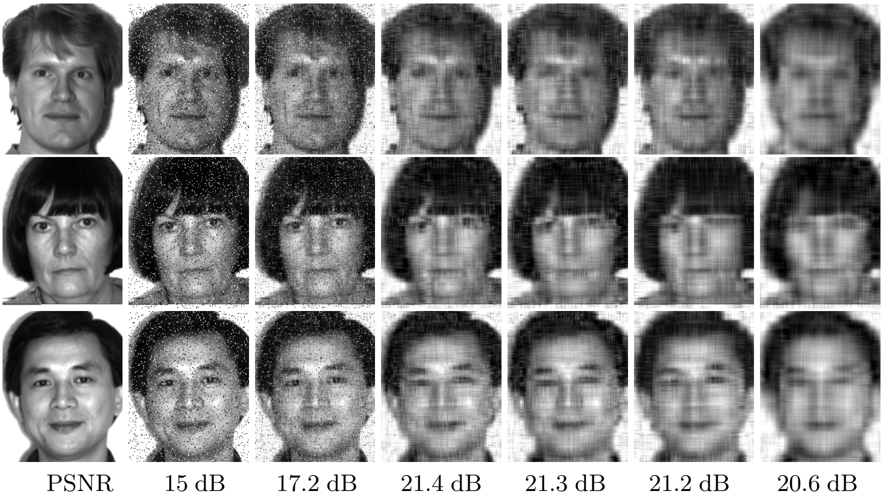
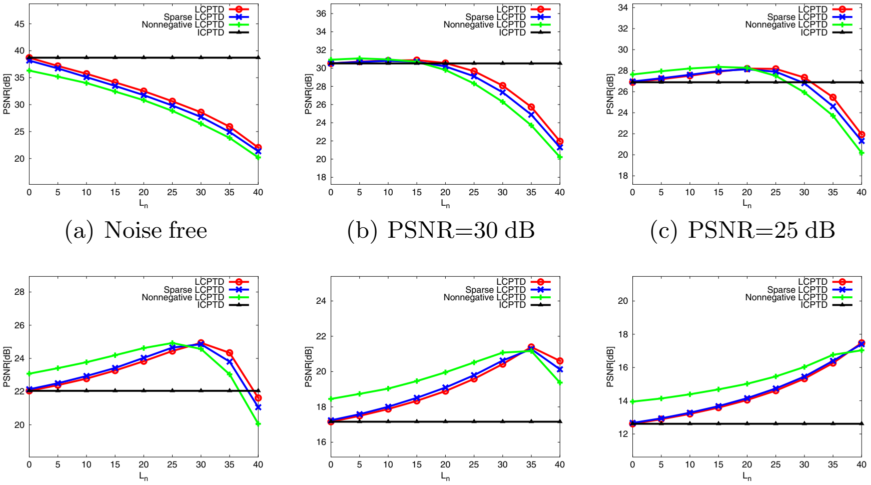


Fig. 2. Face images corrupted by additive noise and the reconstructed images (PSNR= 15 dB, $J = 40$): 1st column: original images, 2nd column: noisy images, 3rd column: ICPTD model, 4th column: LCPTD ($L_n = 35$), 5th column: LCPTD with sparse constraint ($L_n = 35$), 6th column: LCPTD with nonnegative constraint ($L_n = 35$), 7th column: SCPTD model.



(a) Noise free

(b) PSNR=30 dB

(c) PSNR=25 dB

(d) PSNR=20 dB

(e) PSNR=15 dB

(f) PSNR=10 dB

Fig. 3. PSNRs for various noisy data sets

bases) from only two observed tensors. We should note that the selection of L_n could be very important deciding factor to obtain proper decomposition for the LCPTD method from this experiment.

4.2 Images (Faces) Reconstruction and Denoising

In this part, the LCPTD was applied to face reconstruction problems and the performances were compared with other models. The Yale face database consists of 165 gray-scale images of 15 individuals. There are 11 images per subject with different facial expressions or configurations. In this experiments, we used 15 full-face images; we took a one image from each subject. Size of images are 215×171 pixels, then we considered that $I_1 = 215$, $I_2 = 171$, and $S = 15$. Furthermore, we prepared salt-and-pepper noised data sets: SNR $\in \{5,\ 10,\ 20\}$ [dB].

We applied our new LCPTD model, a sparse LCPTD, and a nonnegative LCPTD with various numbers of common components for the noise free and noised data sets; the number of bases was fixed as $J = 40$, and number of common components was changed in $L_n \in \{0,\ 5,\ 10,\ \ldots,\ 40\}$. We computed the PSNR between the original faces and the reconstructed faces.

Fig. 2 depicts the results of face reconstruction. We can see that the ICPTD model couldn't reduce the noise well, and the SCPTD model was robust with respect to noise but reconstructed faces were too fuzzy (distorted). However, the LCPTD based methods gave the appropriate and intermediate decompositions for both models.

Fig. 3 depicts the graphs of PSNR for various number of common components. We can see that if the noise level becomes larger, the maximum points of PSNR move to right. In noise free data set Fig. 3(a), the maximum PSNR was obtained at $L_n = 0$ for all methods; so the ICPTD model is the best for them in this case. On the other hand, the maximum PSNRs were obtained by the LCPTD based methods in Fig. 3(b,c,d,e). It is also interesting that the nonnegative LCPTD kept high PSNR for smaller number of common components in comparison with the other methods in high noise level (see Fig. 3(d,e,f)). It can be considered that the nonnegative constraint is useful for this problem.

In general, because real data includes often some noise factors, the proposed method could be very useful and practical for the real tensor data analysis. The higher noise level requires large number of common components, but multitude of common components often only hampers the fitting. However, we have to select the best parameter of L_n and the open problem is how to select it. We may be able to select L_n depending on PSNR if it is known as prior information.

5 Conclusion

We have presented a method of linked CP tensor decomposition (LCPTD) including sparse and nonnegative factorization by using the HALS algorithm. LCPTD can be considered as a generalized model of simultaneous CP tensor decomposition with common factors, and it provides some improvement over existing methods by selecting optimal parameters of L_n and ξ_n. The parameter selection can be considered as a key issue of flexible model. The Bayesian method or cross validation method may be able to determine such parameters. Its detail and application are reserved for our future works.

References

1. Bro, R.: Multi-way analysis in the food industry - models, algorithms, and applications. Tech. rep., MRI, EPG and EMA, Proc. ICSLP 2000 (1998)
2. Carroll, J., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. Psychometrika 35, 283–319 (1970)
3. Cichocki, A.: Tensor decompositions: New concepts for brain data analysis? Journal of Control, Measurement, and System Integration (SICE) 7, 507–517 (2011)
4. Cichocki, A., Phan, A.: Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences E92-A (3), 708–721 (2009)
5. Cichocki, A., Zdunek, R., Amari, S.-I.: Hierarchical ALS Algorithms for Nonnegative Matrix and 3D Tensor Factorization. In: Davies, M.E., James, C.J., Abdallah, S.A., Plumbley, M.D. (eds.) ICA 2007. LNCS, vol. 4666, pp. 169–176. Springer, Heidelberg (2007)
6. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. Wiley Publishing (2009)
7. Guo, Y., Pagnoni, G.: A unified framework for group independent component analysis for multi-subject fMRI data. NeuroImage 42(3), 1078–1093 (2008)
8. Harshman, R.: Foundations of the parafac procedure: Model and conditions for an 'explanatory' multi-mode factor analysis. UCLA Working Papers in phonetics 16, 1–84 (1970)
9. Jolliffe, I.T., Trendafilov, N.T., Uddin, M.: A modified principal component technique based on the LASSO. Journal of Computational and Graphical Statistics 12(3), 531–547 (2003)
10. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–791 (1999)
11. Lee, H., Choi, S.: Group nonnegative matrix factorization for EEG classification. Journal of Machine Learning Research - Proceedings Track 5, 320–327 (2009)
12. Phan, A., Cichocki, A.: Tensor decompositions for feature extraction and classification of high dimensional datasets. IEICE, NOLTA 1(1), 37–68 (2010)

A Human-Simulated Immune Evolutionary Computation Approach

Gang Xie[1,2], Hong-Bo Guo[2], Yu-Chu Tian[1], and Maolin Tang[1]

[1] School of Elec Eng and Computer Science, Queensland University of Technology,
GPO Box 2434, Brisbane QLD 4001, Australia
{y.tian,m.tang}@qut.edu.au

[2] College of Information Engineering, Taiyuan University of Technology,
79 Yingze West Street, Taiyuan, Shanxi 030024, P.R. China
xiegang@tyut.edu.cn

Abstract. Premature convergence to local optimal solutions is one of the main difficulties when using evolutionary algorithms in real-world optimization problems. To prevent premature convergence and degeneration phenomenon, this paper proposes a new optimization computation approach, human-simulated immune evolutionary algorithm (HSIEA). Considering that the premature convergence problem is due to the lack of diversity in the population, the HSIEA employs the clonal selection principle of artificial immune system theory to preserve the diversity of solutions for the search process. Mathematical descriptions and procedures of the HSIEA are given, and four new evolutionary operators are formulated which are clone, variation, recombination, and selection. Two benchmark optimization functions are investigated to demonstrate the effectiveness of the proposed HSIEA.

Keywords: Human-simulated intelligence, Artificial immune systems, Evolutionary algorithm, Clonal selection, Evolutionary operators.

1 Introduction

Evolutionary algorithms (EAs) are one of the important approaches in stochastic search techniques with the essential characteristics of parallelism, adaptiveness and randomicity. However, there are still challenging difficulties when applying EAs to large-scale and complex real-world optimization problems. One of such difficulties is premature convergence, which occurs when the population reaches a suboptimal state on which most of the operations are no longer functional to produce improved offspring [1,2].

Much effort has been made to improve the performance of EAs. Cen [3] and Yang *et al.* [4] have proposed a hybrid scheme, in which simulated annealing is employed to help an adaptive genetic algorithm escape from local optima and thus prevent premature convergence. Meanwhile, the tabu search algorithm was introduced to increase convergence speed. Herrera *et al.* [1] presented gradual distributed real-coded genetic algorithms that apply a different crossover operator to each subpopulation to deal with the premature problem. Using the

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 92–99, 2012.

concept of information theory, Yeh and Jang [5] and Bhattacharya [6] developed information-guided evolutionary operators to avoid premature convergence. Other developments in this area include references [7] and [8].

Among the developments of various EAs, the Mind Evolutionary Algorithm (MEA) [9,10] was proposed through introducing human-simulated machine learning. It simulates the process of human thinking and learning in certain social environments [11]. In spite of these advances, some shortcomings are also exposed in the applications of the MEA. Because MEA's operators amend the individuals of the population randomly, the degeneration phenomenon becomes inevitable. In particular, when solving a complex real-world problem, the problem's characteristics, which can help resolve the degeneration and improve the convergence speed, are ignored by the MEA.

Artificial immune system technologies are new developments following artificial neural networks and EAs. There have been many successful artificial immune system applications, especially in the optimization area [12,13]. The clonal selection principle is a basic and important model in an artificial immune system. Xie *et al.* [14] incorporated this model into the MEA to deal with the premature convergence problem. However, detailed understanding of the clone selection principle with applications in complex real-world optimization problems are yet to be developed. This motivates the research of this work.

This paper proposes a new optimization computation approach: human-simulated immune evolutionary algorithm (HSIEA). Similar to the MEA, the HSIEA simulates the evolution process of human society and makes use of the co-evolution and information-guided mechanism. However, the HSIEA is fundamentally different from the MEA in algorithm architecture and operators. It will be shown that the HSIEA solves the premature and degeneration problems and outperforms the MEA in computational efficiency.

The paper is organized as follows. Section 2 formalizes the fundamentals of the HSIEA. The flowchart of the HSIEA is developed in section 3. Then, two benchmark functions are investigated to ascertain the good performance of the HSIEA in section 4. Finally, section 5 concludes the paper.

2 Fundamentals of the HSIEA

Investigations into the human intelligence development have revealed that two important and universal modes exist: similar-taxis and dissimilation. The similar-taxis refers to human being's capability of adopting existing technique validated by others to handle various problems; while the dissimilation describes human being's prowess in developing innovative approach from existing ones to deal with unknown fields of the world. These two different modes interact to each other to drive the progress of human intelligence development. During this progress, society division and cooperation are also developed with the understanding that no one will survive and succeed without such an collaborative society environment and the aims of every person's study are definite at the same time. From this understanding, a human-simulated evolutionary computation model can be developed with its mechanism being illustrated in Fig. 1.

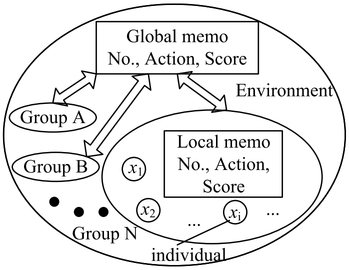


Fig. 1. The Mechanism of the human-simulated evolutionary algorithm

As a multi-group-based evolutionary algorithm, the HSIEA applies the **similar-taxis searching scheme** to achieve the local optimal competition. Two types of similar-taxis searching processes have been embedded into the HSIEA: individual similar-taxis and group similar-taxis.

In individual similar-taxis searching, an individual becomes the winner in a group through local competition, and the winner's information is recorded in the local memos. This process is executed repeatedly, producing a local optimal solution for each group.

In group similar-taxis searching, all groups exchange information for replenishing knowledge that cannot be achieved by any group itself. Also, the global memos will determine the parameter spaces and the number of iterations for every group in the next iteration. Group similar-taxis will be executed when individual similar-taxis meets the terminal condition.

Dissimilation searching is a searching process in which the global solution is selected from the local optimal solutions produced in the similar-taxis searching. Along with the operation of the similar-taxis, some individuals produce several temporary groups in course of searching the whole solution space. If the scores of any temporary group are higher than those of any mature superior group, the temporary group would replace the superior group and become a new superior group. Thus, dissimilation searching is a global competition process.

Mathematically, the HSIEA is formulated as

$$\begin{aligned} HSIEA = \{&\Phi, X, M, N, K, f(X), D(X_i, X_j), \\ &O((T_C, P_C), (T_V, P_V), (T_R, P_R), (T_S, P_S)), E\}, \end{aligned} \tag{1}$$

where Φ is the antigen, i.e., the optimized function for the numerical optimization problem; X represents the solution space of the optimized function and mathematically is the whole of the antibodies set $\{Ab_i(t)\}$, $Ab_i(t)$ indicates the t^{th} time individual; $M \in I$ is the number of initial antibodies (candidates of solutions); $N \in I$ is the number of groups with the highest affinity between the antigen and antibody; K is the number of antibodies in each groups; $f(X)$ denotes the affinity between the antigen and antibody; $D(X_i, X_j)$ is the affinity between antibodies X_i and X_j; O is the operators of the HSIEA; and E is the terminal criterion; respectively. The four operators of the HSIEA are denoted by (T_C, P_C) for clone operator, (T_V, P_V) for variation operator, (T_R, P_R) for recombination operator, and (T_S, P_S) for selection operator, respectively.

Definition 1. ***Antigen-antibody affinity*** *denoted by $f(X)$ is defined as a calculating result after an antibody is substituted into the antigen Φ. It describes the matching degree of the optimal solution to the object function.*

Definition 2. ***Antibody-antibody affinity*** *$D(X_i, X_j)$ is a norm between two affinities when antibodies X_i and X_j are substituted into the antigen Φ:*

$$D(X_i, X_j) = \|f(X_i) - f(X_j)\|, \tag{2}$$

where, $\| \cdot \|$ represents any norm.

Four evolutionary operators of the HSIEA, i.e., the clone operator, variation operator, recombination operator, and selection operator, are respectively described in the following four definitions. The symbol T_α indicates the correspondingly mapping of respective operators, the subscript α denotes the operators, t is the time of iteration, Ab means antibody, P_α is a probability.

Definition 3. ***The clone operator*** *(T_C, P_C) is defined as:*

$$T_C(X) = [T_C(Ab_i(t)] \, , \ i = P_C \times K, \tag{3}$$

$$P_C = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu(X))^2}{2\sigma^2}} \tag{4}$$

where K is the size of an antibody group, μ is the expectation of X, and σ is the standard deviation selected from Equation (5):

$$\sigma_i = \begin{cases} 0.1, & \textit{if } \Delta_{Ab_i} \geq 0.1; \\ \eta \Delta_{Ab_i}, & \textit{if } \Delta_{Ab_i} < 0.1, \end{cases} \tag{5}$$

where, Δ_{Ab_i} is the Euclidean distance in the i^{th} dimension between the new winner and the best winner from older generation; $0 < \eta < 1$ is a constant. Then, the generation after the clone operation is called $Ab_i^{'}(t) = T_C(Ab_i(t))$.

Definition 4. ***The variation operator*** *(T_V, P_V) is defined as:*

$$T_V[X] = [T_V(Ab_i^{'}(t))], \ i = P_V \times K, \tag{6}$$

$$P_V = \begin{cases} P_V^{D_H}(1 - P_V)^{(1-D_H)}, & \textit{if } Ab_i^{'}(t) \in Ab_i(t); \\ 0, & \textit{if } Ab_i^{'}(t) \notin Ab_i(t), \end{cases} \tag{7}$$

where $D_H = d(Ab_i^{'}(t), Ab_i^*(t))$ represents the Hamming Distance of two antibodies. The clone variation operation with probability P_V is carried out on the antibodies generated by the clone operation. The generation of the population after the variation operation is expressed by $Ab_i^*(t) = T_V(Ab_i^{'}(t))$.

In order to reserve the information of the original population, the variation operator is only applied to the new antibodies generated by the clone operation.

Definition 5. ***The recombination operator*** *(T_R, P_R) is described as:*

$$T_R(X) = [T_R(Ab_i^*(t))], \ i = P_R \times K, \tag{8}$$

$$P_R = \begin{cases} > 0, \textit{ if same numbers 0, 1 in } Ab_i^*(t) \textit{ and } Ab_i^{\#}(t); \\ = 0, \textit{ else}, \end{cases} \tag{9}$$

where $Ab_i^{\#}(t) = T_R(Ab_i^*(t)) \cup T_V(Ab_i^{'}(t))$ represents the generation after the recombination operation.

Definition 6. ***The selection operator*** (T_S, P_S) *is described as:*

$$T_S(X) = [Ab_i^{\#}(t) \mid \max f(X) \ or \mid \min f(X)] \tag{10}$$

$$P_S = \begin{cases} 1, & if\ f(Ab_i^{\#}(t)) > f(Ab_i(t+1)); \\ \exp(\Delta f/a), & if\ \Delta f \geq 0\ and\ Ab_i^{\#}(t)\ not\ the\ best\ antibody; \\ 0, & if\ \Delta f \geq 0\ and\ Ab_i^{\#}(t)\ is\ the\ best\ antibody, \end{cases} \tag{11}$$

where $\Delta f = f(Ab_i(t+1)) - f(Ab_i^{\#}(t))$, $a > 0$ *is a value related to the diversity of the antibody population. higher the diversity is, the higher the value of* a *is.*

Definition 7. ***The terminal criterion*** E *is quantitatively described by a limited number of iterations or the best solution that cannot be improved in a certain number of iterations, or a combination of both. A termination criterion can be:*

$$\mid f^* - f^{best} \mid < \varepsilon; \quad OR\!: \ \mid f^* - f^{best} \mid < \varepsilon \mid f^* \mid, \ if\, 0 < \mid f^* \mid < 1, \tag{12}$$

where f^* *is the optimal value of the objective function;* f^{best} *is the best value of the objective function in the current generation.*

3 Logic Flow of the HSIEA

From the fundamentals described in the previous section, the logic flow and procedures of the HSIEA can be developed and are shown in Figure 2.

4 Numerical Experimentation

Two benchmark test functions are investigated in this section to demonstrate the HSIEA: Michalewicz's function denoted by f_1 and the rotated hyper-ellipsoid function denoted by f_2:

$$f_1 = \sum_{i=1}^{5} \sin(x_i) \sin\left(\frac{i \cdot x_i^2}{\pi}\right)^{20}, \ x_i \in [0, \pi], \tag{13}$$

$$f_2 = -\sum_{i=1}^{5} \left(\sum_{j=1}^{i} x_j\right)^2, \ x \in [-65.536, 65.536]. \tag{14}$$

For the Michalewicz's function f_1, there are 5! local optima and the global minimum is $f_{1min} = -4.687$. The rotated hyper-ellipsoid function f_2 has a global minimum $f_{2min} = 0$ at $x_i = 0$, $i = 1, \cdots, 5$.

To make comparisons between the HSIEA and MEA, we consider the convergence speed, the quality of the solution, and the off-line performance [15]: $X_e^*(A) = \frac{1}{T}\sum_{t=1}^{T} f_e^*(Ab_i(t))$, where $f_e^*(Ab_i(t)) = \text{best}\{f_e(Ab_1(t)),\ f_e(Ab_2(t)), \cdots, f_e(Ab_i(t))\}$ is the best object function value or the best affinity at the t^{th} iteration, T is the number of iterations of the algorithm.

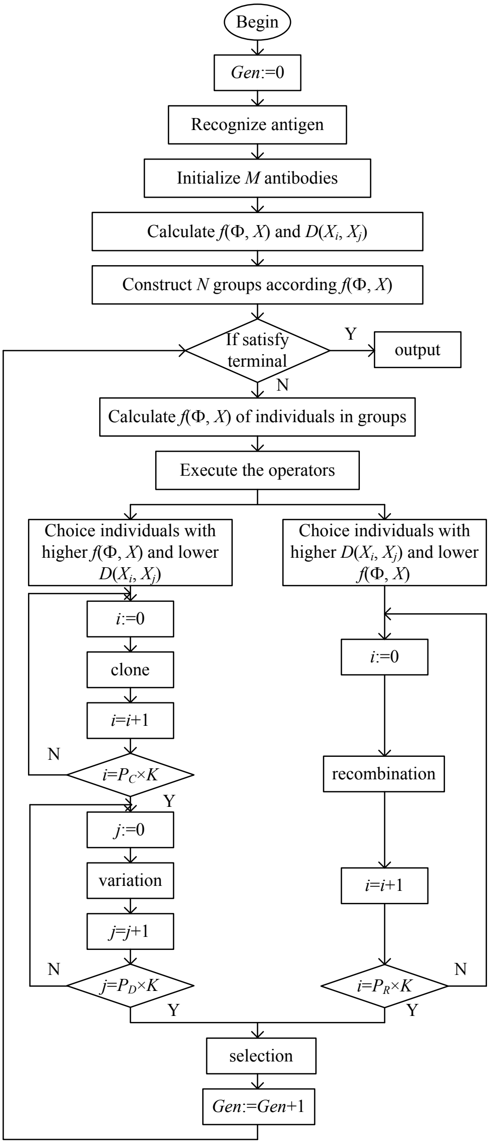

Begin
Gen:=0
Recognize antigen
Initialize M antibodies
Calculate f(Φ, X) and D(X_i, X_j)
Construct N groups according f(Φ, X)
If satisfy terminal
Y
output
N
Calculate f(Φ, X) of individuals in groups
Execute the operators
Choice individuals with higher f(Φ, X) and lower D(X_i, X_j)
Choice individuals with higher D(X_i, X_j) and lower f(Φ, X)
i:=0
clone
i=i+1
N
i=P_C×K
Y
j:=0
variation
j=j+1
N
j=P_D×K
Y
i:=0
recombination
i=i+1
i=P_R×K
N
Y
selection
Gen:=Gen+1

Fig. 2. Flowchart of the HSIEA

In our simulations, the initial number of individuals is set to be $M = 200$. The terminal number of iterations is 100 generations, and the terminal threshold $\varepsilon = 0.0001$. The number of successful optimizations is denoted by N_{TS}, and the number of failures is denoted by N_{TF}, respectively. We have $N_{TS} + N_{TF} = 100$.

The results are summarized in Table 1 and Figures 3 and 4. It is seen from these results that compared with the MEA, the HSIEA not only converges faster but also gives a better solution and off-line performance for both functions.

Table 1. Results of the HSIEA and MEA (the threshold $\epsilon = 0.0001$)

Test function	MEA			HSIEA			Real Solution
	N_{TS}	N_{TF}	Solution	N_{TS}	N_{TF}	Solution	
f_1 in (13)	0	100	−4.583	86	14	−4.679	−4.687
f_2 in (14)	0	100	1.296E+1	97	3	4.979E-10	0

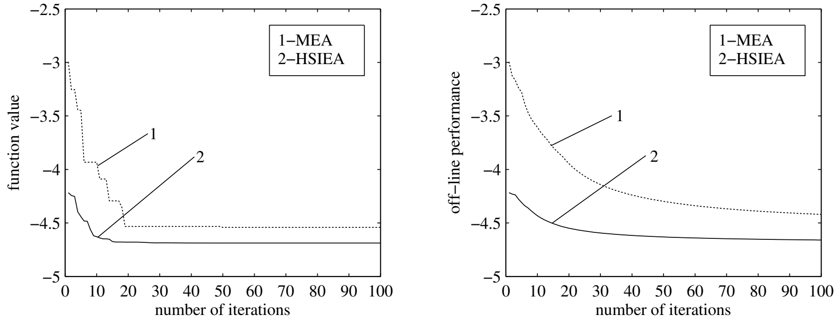


Fig. 3. Optimization of Michalewicz's function in Equation (13)

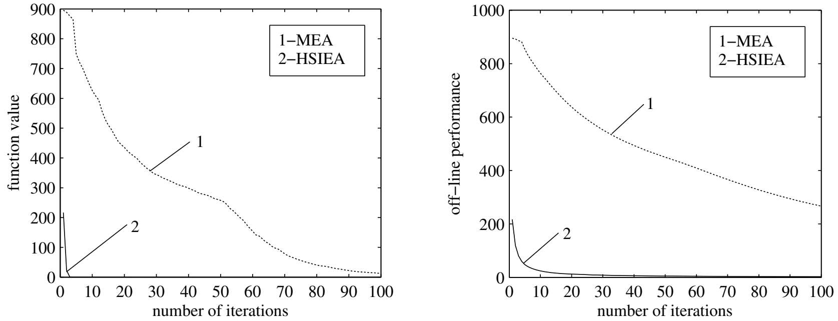


Fig. 4. Optimization of the rotated hyper-ellipsoid function in Equation (14)

5 Conclusion

A new evolutionary algorithm, the HSIEA, has been developed in this paper. The algorithm inherits the advantages of the MEA method and also introduces the features of the artificial immune systems. Because of the introduction of the clonal selection principle, the HSIEA has used several new evolutionary operations such as antigen recognition, clone, variation, recombination, and selection in comparison with the MEA method. This makes the HSIEA fundamentally different from the MEA and other evolutionary algorithms. The effectiveness of the HSIEA approach has been demonstrated through three benchmark functions.

Acknowledgments. Author G. Xie would like to thank the National Natural Science Foundation of China for its support under Grant Number 60975032.

References

1. Herrera, F., Lozano, M.: Gradual Distributed Real-Coded Genetic Algorithms. IEEE Trans. on Evolutionary Computation 4, 43–63 (2000)
2. Paszkowicz, W.: Properties of a Genetic Algorithm Extended by a Random Self-Learning Operator and Asymmetric Mutations: A Convergence Study for a Task of Powder-Pattern Indexing. Analytica Chimica Acta 566, 81–98 (2006)
3. Cen, L.: A Hybrid Genetic Algorithm for the Design of FIR Filters with SPoT Coefficients. Signal Processing 87, 528–540 (2007)
4. Yang, Z., Tian, Z., Yuan, Z.X.: GSA-based Maximum Likelihood Estimation for Threshold Vector Error Correction Model. Computational Statistics and Data Analysis 52, 109–120 (2007)
5. Yeh, C.-W., Jang, S.-S.: The Development of Information Guided Evolution Algorithm for Global Optimization. Journal of Globle Optimization 36, 517–535 (2006)
6. Bhattacharya, M.: Exploiting Landscape Information to Avoid Premature Convergence in Evolutionary Search. In: Proc. IEEE Congress on Evolutionary Computation, CEC 2006, pp. 560–564. IEEE Press, New York (2006)
7. Liang, C.H., Chung, C.Y., Wong, K.P., Duan, X.Z.: Parallel Optimal Reactive Power Flow Based on Cooperative Co-evolutionary Differential Evolution and Power System Decomposition. IEEE Trans. on Power Systems 22, 249–257 (2007)
8. Jiao, L.C., Liu, J., Zhong, W.C.: An Organizational Coevolutionary Algorithm for Classification. IEEE Trans. on Evolutionary Computation 10, 67–80 (2006)
9. Wang, C., Xie, K.: Convergence of a New Evolutionary Computation Algorithm in Continuous State Space. Int. J. Computer Math. 79, 27–37 (2002)
10. Xie, K., Qiu, Y., Xie, G.: Convergence Analysis of Mind Evolutionary Algorithm Based on Functional Analysis. In: Proc. 5th IEEE Int. Conf. on Cognitive Informatics (ICCI 2006), vol. 2, pp. 707–710. IEEE Press, New York (2006)
11. Jie, J., Zeng, J.C., Han, C.Z.: An Extended Mind Evolutionary Computation Model for Optimizations. Appl. Math. and Computation 185, 1038–1049 (2007)
12. Hart, E., Timmis, J.: Application Areas of AIS: The Past, the Present and the Future. Applied Soft Computing 8, 191–201 (2008)
13. Yuan, S.F., Chu, F.L.: Fault Diagnosis Based on Support Vector Machines with Parameter Optimisation by Artificial Immunisation Algorithm. Mechanical Systems and Signal Processing 21, 1318–1330 (2007)
14. Xie, G., Xu, X.Y., Xie, K.N., Chen, Z.H.: Clone Mind Evolution Algorithm. In: Wang, L., Chen, K., Ong, Y.S. (eds.) ICNC 2005. LNCS, vol. 3611, pp. 945–950. Springer, Heidelberg (2005)
15. Digalakis, J.G., Margaritis, K.G.: An Experimental Study of Benchmarking Functions for Genetic Algorithms. Int. J. Computer Math. 79, 403–416 (2002)

A STPHD-Based Multi-sensor Fusion Method

Lu Zhenwei, Zhao Lingling, Su Xiaohong, and Ma Peijun

PosBox 319, Harbin Institute of Technology, No.92, West Da-zhi Street, Harbin, China
luzhenweiwei@163.com, zhaolinglinghit@126.com,
{sxh,ma}@hit.edu.cn

Abstract. In order to extract the peaks of PHD, a novel method STPHD has been proposed recently. This method can provide more accurate target state estimates than the general clustering algorithm such as k-means clustering. This paper presents a version of STPHD for multi-sensor scene and makes two contributions. First, we generalize the STPHD algorithm to a multi-sensor scenario with an existing framework of fusion. The framework includes an association step and a fusion step. This generation can get better performance in accuracy. But the association step is time-consuming. The second contribution is a novel model for computing the cost of two sets of particles with sub-weights in the association step. The numerical simulation results show that the proposed method can significantly reduce the time cost with a very slight loss in accuracy compared with the previous methods.

Keywords: Multi-sensor fusion, STPHD, Particle filter.

1 Introduction

In recent twenty years, a new method called Probability Hypothesis Density (PHD) filter proposed by Mahler [1] attracts much interest. In contrast to the conventional Bayesion filter, this method propagates the first moment of posterior density, so called PHD, in fact an intensity function instead of the probability density function (pdf) which takes a lot in computation. The advantage of the PHD filter is that the expected number of targets can be obtained by computing the integral on the region of interest while the disadvantage is that there is no analytic solution to the filter. The mainly implementations of PHD are the Sequential Monte Carlo (SMC) method [2] and the Gaussian Mixture (GM) method [3], and the latter developed in the linear/Gaussian dynamics can obtain a closed-form solution while the former with dense computing task can be used widely not only the linear/Gaussian condition. The subject we discuss in this paper is based on the SMC method.

A phase of extracting state is necessary after estimating the number of targets. The method of clustering analysis, such as k-means clustering [7] and Tobias' peak extraction algorithm [8], is used in this step. The drawback of the clustering is no particle can be assigned to multiple targets as every particle has one weight. A novel approach called STPHD (Single-Target PHD) is proposed by Zhao in [4], which has been verified by simulation that it outperforms the above methods. In the new method, a vector

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 100–107, 2012.

of weights is attached to a particle, each component of which responds to a specific measure.

In this paper, we propose a generalization of the STPHD in the multi-sensor scenario based on an existing framework [5] and give a new method of computing the cost of two set of weighted particles. Simulations are also provided to measure the performance of the fusion algorithm. The remainder of the paper is organized as follows. In section 2, we review the SMC implementation of PHD. A brief description of STPHD will be presented in section 3. We introduce a framework of association and fusion of multi-sensor in section 4. The results of simulation are provided in section 5 while the conclusion is in section 6.

2 The SMC Implementation of PHD

Due to the high dimensionality of integral, there isn't analytic solution to the PHD equations in a general condition. To make the method feasible, a particle filter implementation can be introduced into PHD to get a suboptimal solution. The brief three steps will be described as follows.

2.1 Prediction

In prediction step, we still assume that the set of weighted particles $\left\{x_{k-1}^i, w_{k-1}^i\right\}_{i=1}^{L_{k-1}}$ at time $k-1$ is available. For $i=1,\dots,L_{k-1}$, sample $\tilde{x}_k^i$ from $q_k\left(\cdot \mid x_{k-1}^i, Z_k\right)$ responding to the survival particles and evaluate the predicted weights. For $i=L_{k-1}+1,\dots,L_{k-1}+J_k$, where J_k denotes the number of the particles new born, sample $\tilde{x}_k^i$ from $p_k\left(\cdot \mid Z_k\right)$ responding to the new born particles and evaluate the predicted weights. Thus we get the predicted set of weighted particles $\left\{\tilde{x}_{k-1}^i, \tilde{w}_{k|k-1}^i\right\}_{i=1}^{L_{k-1}+J_k}$.

2.2 Correction

In correction step, the weights can be updated by the collection of measures arriving. For $i=1,\dots,L_{k-1}+J_k$, the new weights are evaluated by

$$\tilde{w}_k^i = [\upsilon\left(\tilde{x}_k^i\right) + \sum_{z_{k,j}\in Z_k} \frac{\psi_{k,z_{k,j}}\left(\tilde{x}_k^i\right)}{\kappa_k\left(z_{k,j}\right) + C_k\left(z_{k,j}\right)}]\tilde{w}_{k|k-1}^i \tag{1}$$

where $C_k\left(z_{k,j}\right) = \sum_{i=1}^{L_{k-1}+J_k} \psi_{k,z_{k,j}}\left(\tilde{x}_k^i\right)\tilde{w}_{k|k-1}^i$. Thus we get the updated weighted particles $\left\{\tilde{x}_k^i, \tilde{w}_k^i\right\}_{i=1}^{L_{k-1}+J_k}$.

2.3 Resampling

In resampling step, we evaluate the sum of all the weighs by $\hat{N}_{k|k} = \sum_{i=1}^{L_{k-1}+J_k} \tilde{w}_k^i$ as the expected number of the targets in the surveillance region. But $\hat{N}_{k|k}$ may not be an integer, so we use the nearest integer $\hat{T}_k = round\left(\hat{N}_{k|k}\right)$ instead. L_k particles should be resampled from the updated particles set $\left\{\tilde{x}_k^i, \tilde{w}_k^i \big/ \hat{N}_{k|k}\right\}_{i=1}^{L_{k-1}+J_k}$ where $L_k = \hat{T}_k \cdot \rho$, ρ denotes the number of particles assigned to each target. Thus we get the set of weighted particles after resampling $\left\{x_k^i, w_k^i\right\}_{i=1}^{L_k}$ where $w_k^i = \hat{N}_{k|k} \big/ L_k$.

3 STPHD

In [4], Zhao has demonstrated that in the correction step, the updated PHD $D_{k|k}\left(x \mid Z_{1:k}\right)$ can be expressed as

$$D_{k|k}\left(x \mid Z_{1:k}\right) = \sum_{z \in Z_k} \Delta D_{k|k}\left(x \mid z\right) + \Delta D_{k|k}\left(x \mid \phi\right) \tag{2}$$

where $\Delta D_{k|k}\left(x \mid \phi\right) = \upsilon(x) D_{k|k-1}$ denotes the PHD of the measure undetected.

So in correction step of the particles filter implementation of PHD, $G_k^{i,j}$ and $\tilde{w}_k^{i,j}$ should be computed by

$$G_k^{i,j} = \frac{\psi_{k,z_{k,j}}\left(\tilde{x}_k^i\right)}{\kappa_k\left(z_{k,j}\right) + C_k\left(z_{k,j}\right)} \tag{3}$$

and

$$\tilde{w}_k^{i,j} = G_k^{i,j} \tilde{w}_{k|k-1}^i \tag{4}$$

Suppose that the expected number of target we estimate is $\hat{T}_k$, compute the sum of sub-weights responding to $z_{k,j}$ denoted by W_k^j where $W_k^j = \sum_{i=1}^{L_{k-1}+J_k} \tilde{w}_k^{i,j}$. Then choose the greatest $\hat{T}_k$ ones to estimate the target state.

4 Multi-sensor Association Fusion

A framework of multi-sensor particle filter cloud fusion was proposed in [5], which consists of association step and fusion step. These steps will be presented as follows. Like literature [5], we assume there are only two sensors in the surveillance region and we only discuss the condition at time k , so we ignore the index k in the following discussions.

4.1 Association Step

Suppose that $\hat{T}_{(1)}$ targets are estimated from sensor 1 and $\hat{T}_{(2)}$ targets are estimated from sensor 2. From the STPHD, we obtain the responding two sets of particles with sub-weight $\left\{\left\{\tilde{x}_{(1)}^{i_1}, \tilde{w}_{(1)}^{i_1,j_1}\right\}_{i_1=1}^{L_{(1)}}\right\}_{j_1=1}^{\hat{T}_{(1)}}$ and $\left\{\left\{\tilde{x}_{(2)}^{i_2}, \tilde{w}_{(2)}^{i_2,j_2}\right\}_{i_2=1}^{L_{(2)}}\right\}_{j_2=1}^{\hat{T}_{(2)}}$, where $L_{(1)}$ and $L_{(2)}$ denote the number of particles at the two sensors respectively. The notations $\Xi_{(1)}^{j_1}$ and $\Xi_{(2)}^{j_2}$ are used to replace $\left\{\tilde{x}_{(1)}^{i_1}, w_{(1)}^{i_1,j_1}\right\}_{i_1=1}^{L_{(1)}}$ and $\left\{\tilde{x}_{(2)}^{i_2}, w_{(2)}^{i_2,j_2}\right\}_{i_2=1}^{L_{(2)}}$ where $w_{(1)}^{i_1,j_1} = \tilde{w}_{(1)}^{i_1,j_1} / \sum_{i_1=1}^{L_{(1)}} \tilde{w}_{(1)}^{i_1,j_1}$ and $w_{(2)}^{i_2,j_2} = \tilde{w}_{(2)}^{i_2,j_2} / \sum_{i_2=1}^{L_{(2)}} \tilde{w}_{(2)}^{i_2,j_2}$.

Thus the problem of association can be handled by solving the following constraint condition

$$\min_{\chi} \sum_{j_1=1}^{\hat{T}_{(1)}} \sum_{j_2=1}^{\hat{T}_{(2)}} \chi_{j_1 j_2} \operatorname{cost}\left(\Xi_{(1)}^{j_1}, \Xi_{(2)}^{j_2}\right) \tag{5}$$

subject to

$$\sum_{j_1=1}^{\hat{T}_{(1)}} \chi_{j_1 j_2} = 1, j_2 = 1, \ldots, \hat{T}_{(2)}$$

$$\sum_{j_2=1}^{\hat{T}_{(2)}} \chi_{j_1 j_2} = 1, j_1 = 1, \ldots, \hat{T}_{(1)}$$

where χ is a matrix and the binary variable $\chi_{j_1 j_2}$ is its element at j_1 th row and j_2 th column, $\chi_{j_1 j_2} \in \{0,1\}$. $\chi_{j_1 j_2} = 1$ means that the j_1 th set of particles and the j_2 th set of particles have a associated relation. $\chi_{j_1 j_2} = 0$ means they have no relation. Suppose $\Xi_{(1)}^{j_1}$ and $\Xi_{(2)}^{j_2}$ attach to the same target. We denote the cost of the hypothesis by $\operatorname{cost}\left(\Xi_{(1)}^{j_1}, \Xi_{(2)}^{j_2}\right)$. It can be evaluated by

$$\operatorname{cost}\left(\Xi_{(1)}^{j_1}, \Xi_{(2)}^{j_2}\right) = \min \sqrt[p]{\sum_{i_1=1}^{L_{(1)}} \sum_{i_2=1}^{L_{(2)}} w_{i_1 i_2} d\left(\tilde{x}_{(1)}^{i_1}, \tilde{x}_{(2)}^{i_2}\right)} \tag{6}$$

$$\sum_{i_1=1}^{L_{(1)}} w_{i_1 i_2} = w_{(2)}^{i_2,j_2}, i_2 = 1, \ldots, L_{(2)}$$

$$\sum_{i_2=1}^{L_{(2)}} w_{i_1 i_2} = w_{(1)}^{i_1,j_1}, i_1 = 1, \ldots, L_{(1)}$$

$$w_{i_1 i_2} \geq 0, i_1 = 1, \ldots, L_{(1)}, i_2 = 1, \ldots, L_{(2)}$$

where $d\left(\tilde{x}_{(1)}^{i_1}, \tilde{x}_{(2)}^{i_2}\right)$ is the distance between $\tilde{x}_{(1)}^{i_1}$ and $\tilde{x}_{(2)}^{i_2}$. The minimization above can be solved by using linear programming interior point methods in matlab. But it takes too much time. So we provide another form of $\mathrm{cost}\left(\Xi_{(1)}^{j_1}, \Xi_{(2)}^{j_2}\right)$ by

$$\mathrm{cost}\left(\Xi_{(1)}^{j_1}, \Xi_{(2)}^{j_2}\right) = \min \sqrt[p]{\sum_{i_1=1}^{L_{(1)}} \sum_{i_2=1}^{L_{(2)}} w_{i_1 i_2} d\left(w_{(1)}^{i_1, j_1} \tilde{x}_{(1)}^{i_1}, w_{(2)}^{i_2, j_2} \tilde{x}_{(2)}^{i_2}\right)} \tag{7}$$

$$\sum_{i_1=1}^{L_{(1)}} w_{i_1 i_2} = \frac{1}{L_{(2)}}, i_2 = 1, \ldots, L_{(2)}$$

$$\sum_{i_2=1}^{L_{(2)}} w_{i_1 i_2} = \frac{1}{L_{(1)}}, i_1 = 1, \ldots, L_{(1)}$$

$$w_{i_1 i_2} \geq 0, i_1 = 1, \ldots, L_{(1)}, i_2 = 1, \ldots, L_{(2)}$$

For a weighted particle, the weight presenting the importance is an important part of the particle. When we estimate the state of a target from a set of weighted particles, we should evaluate the weighted sum of particles with normalized weights. Similarly, we use the weight $w_{(s)}^{i_s, j_s}$ to multiply the state vector of particle $\tilde{x}_{(s)}^{i_s}, s = 1,2$ to get a new state vector of particle. We consider the new ones have no difference in the region of weight since the information of weight has been fused into the new state vector. Every new state vector of particle includes the previous state vector and previous weight information. So the group of new state vectors can be considered as a sequence of characters without order of the previous set of weighted particles. The difference between the two sets of particles can be measured by the distance of two groups of characters. Thus, we get the new minimization (7), which is used to alternate the previous minimization (6). To solve the new minimization problem above, the Munkres algorithm in [8] can be used. Thus, due to the best χ, we get some associated pairs such as $\left\{\Xi_{(1)}^{l_1}, \Xi_{(2)}^{l_2}\right\}$.

4.2 Fusion Step

Now we have $\Xi_{(1)}^{l_1} = \left\{\tilde{x}_{(1)}^{i_1}, w_{(1)}^{i_1, l_1}\right\}_{i_1=1}^{L_{(1)}}$ and $\Xi_{(2)}^{l_2} = \left\{\tilde{x}_{(2)}^{i_2}, w_{(2)}^{i_2, l_2}\right\}_{i_2=1}^{L_{(2)}}$. To fuse the two sets of weighted particles, a fusion method with covariance in [7] can be described as:

$$\zeta = \left(P_1^{-1} + P_2^{-1}\right)^{-1} \left(P_1^{-1} \sum_{i_1=1}^{L_{(1)}} w_{(1)}^{i_1, l_1} \tilde{x}_{(1)}^{i_1} + P_2^{-1} \sum_{i_2=1}^{L_{(2)}} w_{(1)}^{i_1, l_1} \tilde{x}_{(2)}^{i_2} \right) \tag{8}$$

$$P_s = \sum_{i_s=1}^{L_{(s)}} w_{(s)}^{i_s, j_s} \left(\tilde{x}_{(s)}^{i_s} - \bar{x}_{(s)}\right)\left(\tilde{x}_{(s)}^{i_s} - \bar{x}_{(s)}\right)^T \tag{9}$$

$$\bar{x}_{(s)} = \sum_{i_s=1}^{L_{(s)}} w_{(s)}^{i_s, j_s} \tilde{x}_{(s)}^{i_s}, s = 1,2 \tag{10}$$

Figure 1 shows the entire fusion process of multi-sensor based on STPHD. In association step, we have two different forms of models i.e. the previous minimization (6) and the new minimization (7).

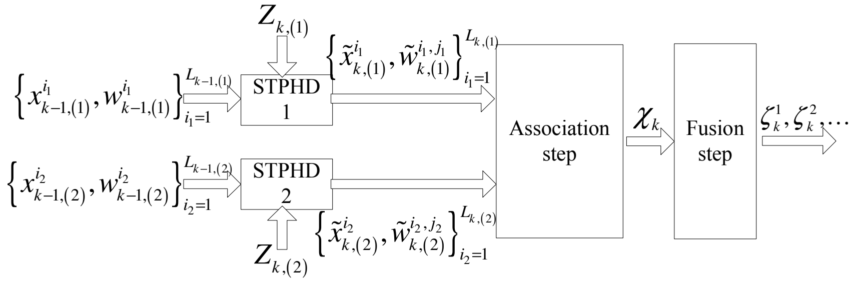

$Z_{k,(1)}$
$\left\{x_{k-1,(1)}^{i_1}, w_{k-1,(1)}^{i_1}\right\}_{i_1=1}^{L_{k-1,(1)}}$
STPHD 1
$\left\{\tilde{x}_{k,(1)}^{i_1}, \tilde{w}_{k,(1)}^{i_1,j_1}\right\}_{i_1=1}^{L_{k,(1)}}$
$\left\{x_{k-1,(2)}^{i_2}, w_{k-1,(2)}^{i_2}\right\}_{i_2=1}^{L_{k-1,(2)}}$
STPHD 2
$\left\{\tilde{x}_{k,(2)}^{i_2}, \tilde{w}_{k,(2)}^{i_2,j_2}\right\}_{i_2=1}^{L_{k,(2)}}$
$Z_{k,(2)}$
Association step
χ_k
Fusion step
$\zeta_k^1, \zeta_k^2, \ldots$

Fig. 1. The entire process of the fusion of multi-sensor based on STPHD

5 Simulations

We assume the number of sensors is two as [5], and the positions are $\left(x^{(1)}, y^{(1)}\right) = (0,-100)$ and $\left(x^{(2)}, y^{(2)}\right) = (0,100)$. Suppose that the targets number is unknown time-varying and the targets move in the region $[-100,100] \times [-100,100]$. The state equation is

$$X_k = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix} X_{k-1} + \begin{pmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{pmatrix} v_{k-1} \tag{11}$$

and the measure equation is

$$r_k^{(s)} = \left\| \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} X_k - \begin{pmatrix} x^{(s)} \\ y^{(s)} \end{pmatrix} \right\| + n_{1,k}^{(s)} \tag{12}$$

$$\theta_k^{(s)} = \arctan\left(\frac{y_k - y^{(s)}}{x_k - x^{(s)}} \right) + n_{2,k}^{(s)}, s = 1,2 \tag{13}$$

where $X_k = \left(x_k, \dot{x}_k, y_k, \dot{y}_k\right)^T$, v_k is a 2-D zero-mean Gaussian white noise with the covariance matrix $diag[1,0.01]$ and $n_{i,k}^{(j)}$ $(i, j = 1,2)$ are the independent zero-mean Gaussian white noise with standard deviation 2.5. $T = 1$ is the sampling period. There are four targets as [5] appearing randomly and their initial positions follow the intensity function $p_k = N(\cdot, \bar{x}, Q)$ where $\bar{x} = (0,3,0,-3)^T$ and $Q = diag[10,1,10,1]$. The number of clutter follows a Poisson distribution with the parameter γ while the clutter points follow a uniform distribution in the surveillance region. We set the detection probability $P_D = 1$ and the number of scans is 15.

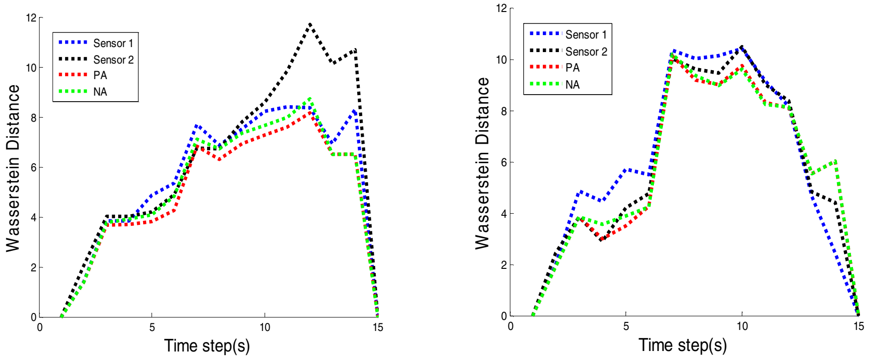


Fig. 2. Wasserstein miss-distance at *r*=0 and *r*=5

We denote the previous algorithm from the equation (6) by PA and denote the new algorithm from (13) by NA, and compare them as follows.

The four Wasserstein miss-distances displayed in figure 5 shows the results of sensor 1, PA and NA. The left one is at *r*=0 while the right one is at *r*=5.

Table 1 and table 2 shows the average results over 20 Monte Carlo runs including the mean and the variance of the miss-distance error and the associated time of PA and NA at *r*=0 and *r*=5 respectively. From the two tables, we can see that PA and NA outperform the algorithm without fusion. And compared with NA, PA has a significant advantage in running time and a comparable accuracy. The NA spends much less time than PA.

Table 1. Average results over 20 Monte Carlo runs (r=0)

	Miss distance error(r=0)		**Run time(s)**
	Mean	**Variance**	
PA	5.208	0.780	48.151
NA	5.461	0.900	0.403
Sensor 1	5.831	1.388	/
Sensor 2	6.528	1.372	/

Table 2. Average results over 20 Monte Carlo runs (r=5)

	Miss distance error(r=5)		**Run time(s)**
	Mean	**Variance**	
PA	6.456	0.792	15.157
NA	6.528	0.798	0.254
Sensor 1	7.010	1.637	/
Sensor 2	6.572	1.069	/

6 Conclusions

In this paper, we generalized the STPHD algorithm to a multi-sensor scenario based on an existing framework and give a new method to compute the cost of two set of weighted particles. The simulation results show that the NA can obviously reduce the time cost with a very little loss in accuracy compared with the PA.

Acknowledgment. This paper was supported by the National Natural Science Foundation of China (NSFC) under Grant 61175027.

References

1. Mahler, R.: Mutlitarget Bayse Fitering via First-order Multitarget Moments. IEEE Trans. Aerosp. Electron. Syst. 39, 1152–1178 (2003)
2. Vo, B., Singh, S., Doucet, A.: Sequential Monte Carlo Methods for Multi-target Filtering with Random Finite Sets. IEEE Trans. Aerosp. Electron. Syst. 41, 1124–1245 (2005)
3. Vo, B., Ma, W.K.: A Closed-Form Solution for the Probability Hypothesis Density Filter. In: 8th International Conference on Information Fusion, Philadelphis, PA, pp. 856–863 (2005)
4. Zhao, L.L., Ma, P.J., Su, X.H., Zhang, H.T.: A New Multi-target State Estimation Algorithm for PHD Particle Filter. In: 13th International Conference on Information Fusion, Edinburgh, UK, pp. 1–8 (2010)
5. Danu, D., Kirubarajan, T., Lang, T., McDonald, M.: Multisensor Particle Filter Cloud Fusion for Multitarget Tracking. In: 11th International Conference on Information Fusion, Cologne, Germany, pp. 1191–1198 (2008)
6. Hoffman, J.R., Mahler, R.P.S.: Multitarget Miss Distance via Optimal Assignment. IEEE Trans. Syst. Man. CY. A. 34, 327–336 (2004)
7. Clark, D.E., Bell, J., Watt, H.: Multi-target State Estimation and Track Continuity for the Particle PHD Filter. IEEE Trans. Aerosp. Electron. Syst. 43(4), 1441–1453 (2007)
8. Tobias, M., Lanterman, A.D.: Techniques for Birth-particle Placement in the Probability Hypothesis Density Particle Filter Applied to Passive Radar. IET Radar Sonar Nav. 2(5), 351–365 (2008)

Group Sparse Inverse Covariance Selection with a Dual Augmented Lagrangian Method

Satoshi Hara and Takashi Washio

The Institute of Scientific and Industrial Research (ISIR), Osaka University, Japan
{hara,washio}@ar.sanken.osaka-u.ac.jp

Abstract. Sparse Inverse Covariance Selection (SICS) is a popular tool identifying an intrinsic relationship between continuous random variables. In this paper, we treat the extension of SICS to the grouped feature model in which the state-of-the-art SICS algorithm is no longer applicable. Such an extended model is essential when we aim to find a groupwise relationships between sets of variables, e.g. unknown interactions between groups of genes. We tackle the problem with a technique called Dual Augmented Lagrangian (DAL) that provides an efficient method for grouped sparse problems. We further improve the DAL framework by combining the Alternating Direction Method of Multipliers (ADMM), which dramatically simplifies the entire procedure of DAL and reduce the computational cost. We also provide empirical comparisons of the proposed DAL–ADMM algorithm against existing methods.

Keywords: Sparse Inverse Covariance Selection, Dual Augmented Lagrangian, Alternating Direction Method of Multipliers.

1 Introduction

The identification of a graphical model structure corresponds to finding a conditional independence among random variables. In a Graphical Gaussian Model where a continuous random variable $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^\top \in \mathbb{R}^d$ follows a zero–mean normal distribution $\mathcal{N}(\mathbf{0}_d, \Lambda^{-1})$, the identification of a structure is equivalent to finding zero entries in a *precision matrix* $\Lambda \in \mathbb{R}^{d\times d}$ or an inverse covariance matrix. This means that two variables x_i and x_j are statistically independent given remaining variables if and only if the (i, j)th entry of Λ is zero, that is,

$$x_i \perp\!\!\!\perp x_j \mid \text{other variables} \quad \Leftrightarrow \quad \Lambda_{ij} = 0\,.$$

Finding zero entries in a precision matrix is referred as *covariance selection* [1]. Recently, several authors have proposed to use a ℓ_1 regularization technique for this problem [2–4]. The problem is formulated as

$$\min_{\Lambda \succ 0} f(\Lambda) \;\equiv\; -\log\det\Lambda + \mathrm{tr}\,(S\Lambda) + \rho\|\Lambda\|_1 \tag{1}$$

where $S \in \mathbb{R}^{d\times d}$ is a sample covariance matrix, $\|\Lambda\|_1$ is an entry-wise ℓ_1-norm defined as $\|\Lambda\|_1 = \sum_{i,j=1}^{d} |\Lambda_{ij}|$ and ρ is a non–negative regularization parameter.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 108–115, 2012.

The positive definiteness of Λ is imposed so that the optimal parameter Λ^* to become a valid precision matrix. We refer to the problem (1) as Sparse Inverse Covariance Selection (SICS) following Scheinberg et al. [5]. Note the case $\rho = 0$ corresponds to the maximum likelihood estimate $\Lambda^* = S^{-1}$. For $\rho > 0$, the additional ℓ_1 term enforces some small entries in Λ to shrink to exactly zeros and the estimator Λ^* gets sparse.

The ℓ_1 regularization term in (1) can be replaced with some other norms on Λ. Duchi et al. [6] have introduced a group structure in the elements of Λ and generalized the problem using group–regularization techniques [7, 8]. This extended group SICS model is helpful when we aim to find the dependency between set of variables, e.g. unknown interactions between groups of genes [6]. Both SICS (1) and group SICS are convex optimization problems and the unique global optimum is available. Especially for the problem (1), several optimization procedures have been proposed [5, 6, 9–12]. Amongst these methods, QUIC [12] would be the most practical state-of-the-art method with some theoretical guarantees. However, the efficiency of QUIC heavily depends on the specific property of the ℓ_1–norm and not applicable to the extended framework in general.

The main scope of this paper is to propose a new algorithm for the group SICS problem [6] which can treat general group regularization terms. The proposed method relies on the Dual Augmented Lagrangian (DAL) method [13] which provides an efficient algorithm for convex and sparse regularization problems. We further update the DAL framework by combining the Alternating Direction Method of Multipliers (ADMM) [5, 10, 14] and propose a DAL–ADMM algorithm. This update makes the entire procedure dramatically simple and helps reducing the practical computational cost.

The remainder of the paper is organized as follows. In Section 2, we review the extended SICS problem with a group structure. In Section 3, we introduce the DAL based optimization method, and then update it by combining ADMM and propose DAL–ADMM algorithm in Section 4. The validity of the proposed method is presented through synthetic experiments in Section 5. Finally, we conclude the paper in Section 6.

2 Group Sparse Inverse Covariance Selection

In this section, we briefly review the extension of SICS (1) into its grouped variant [6, 15]. In group SICS, all d^2 entries in a precision matrix Λ are partitioned into M disjoint groups. Here, let $\mathcal{I}$ be a set of all d^2 indexes in Λ, that is, $\mathcal{I} = \{(i, j); 1 \le i, j \le d\}$. Then, each of M groups $\mathcal{G}_m$ $(1 \le m \le M)$ is represented as a subset of $\mathcal{I}$ where $\mathcal{G}_m \cap \mathcal{G}_{m'} = \phi$ for $m \ne m'$ and $\cup_{m=1}^{M} \mathcal{G}_m = \mathcal{I}$. We also use a notation $\Lambda_{\mathcal{G}_m}$ to represent a vector composed of entries in Λ specified by $\mathcal{G}_m$, that is, $\Lambda_{\mathcal{G}_m} = (\Lambda_{ij})_{(i,j)\in\mathcal{G}_m}$. While the objective of the ordinal SICS is to identify whether each (i, j)th entry of Λ is zero or not, the objective of group SICS is to infer which of $\Lambda_{\mathcal{G}_m}$ gets simultaneously zeros among M groups. For example, this setting is relevant to the identification of dependencies between two sets of genes. In such a case, we partition the entries of Λ into four disjoint groups; two of them corresponds to the block–diagonal entries representing inner group interactions while the other twos specify block–off–diagonal entries related to the interaction between the groups. If latter two entries are simultaneously zeros, it means that two sets of genes do not involve any interactions between them.

Duchi et al. [6] and Schmidt et al. [15] formulated this problem as follows using group–regularization techniques [7, 8],

$$\min_{\Lambda \succ 0} \; g(\Lambda) \;\equiv\; -\log\det\Lambda + \mathrm{tr}\,(S\Lambda) + \sum_{m=1}^{M} \rho_m \|\Lambda_{\mathcal{G}_m}\|_{p_m} \,. \tag{2}$$

Here, $\|\Lambda_{\mathcal{G}_m}\|_{p_m}$ is a ℓ_{p_m}–norm of $\Lambda_{\mathcal{G}_m}$ with $p_m \in [1, \infty]$ and parameters ρ_m and p_m are assigned individually to each group. Note that the problem (2) is a generalization of SICS since setting $\rho_m = \rho$ and $p_m = 1$ results in (1). For $p_m > 1$, a set of parameters $\Lambda_{\mathcal{G}_m}$ shrinks to zeros simultaneously due to the group effect. Hence, the optimal solution Λ^* has a group–wise sparse structure. A parameter p_m is typically set to 2 or ∞ due to computational considerations. In the latter of this paper, we focus on these two specific cases.

3 Dual Augmented Lagrangian for Group SICS

Now, we derive the algorithm for group SICS (2) using Dual Augmented Lagrangian (DAL) [13]. DAL is an algorithm applying an Augmented Lagrangian technique [14] to the dual of the target problem. It is known that DAL is super–linearly convergent, hence it is well suited for sparse regularization problems [13].

The dual of group SICS [6] is given as

$$\min_{W \succ 0, Y} \; -\log\det W \;\text{ s.t. }\; \|Y_{\mathcal{G}_m}\|_{q_m} \le \rho_m \,, \; W - Y - S = 0_{p\times p} \,. \tag{3}$$

Here, $W \in \mathbb{R}^{d\times d}$ is a dual parameter, which satisfies $W^* = \Lambda^{*-1}$ at its optimal from the duality [6]. An operator $\| * \|_{q_m}$ denotes a dual norm of $\| * \|_{p_m}$ with $p_m^{-1} + q_m^{-1} = 1$. We have also introduced the additional parameter Y for the sake of compatibility with the latter discussion. A subscript $\mathcal{G}_m$ is used for Y according as $\Lambda_{\mathcal{G}_m}$. In DAL, we first formulate the following AL function,

$$\mathcal{L}_\beta(W, Y, Z) = -\log\det W + \sum_{m=1}^{M} \delta_{\rho_m}(Y_{\mathcal{G}_m}) + \frac{\beta}{2}\left\|W - Y + \beta^{-1}Z - S\right\|_{\mathrm{F}}^2$$

where $\beta > 0$ is an algorithm parameter, $\| * \|_{\mathrm{F}}$ is a Frobenius–norm, $Z \in \mathbb{R}^{d\times d}$ is a Lagrange multiplier, and $\delta_{\rho_m}(Y_{\mathcal{G}_m})$ is an indicator function given as

$$\delta_{\rho_m}(Y_{\mathcal{G}_m}) = \begin{cases} 0 \;, & \|Y_{\mathcal{G}_m}\|_{q_m} \le \rho_m \\ \infty \;, & \text{otherwise} \end{cases} \;.$$

Note an AL function with $\beta = 0$ corresponds to the ordinal Lagrangian function. Here, we limit ourselves to the case $p_m = q_m = 2$. The basic approach of DAL is to relax the equality constraint of (3) in the intermediate steps of the algorithm and make it fulfilled at the termination. In every DAL updates, we first optimize W and Y so that $\mathcal{L}_\beta(W, Y, Z^{(k)})$ is minimized. Then, we update the dual parameter $Z^{(k)}$ as $Z^{(k+1)} = Z^{(k)} + \beta(W^{(k+1)} - Y^{(k+1)} - S)$. In every steps, a value of β is also gradually increased so that the super–linear convergence is achieved [13]. Under $q_m = 2$, the optimization of Y for a fixed $W^{(k+1)}$ is merely minimizing $\|Y_{\mathcal{G}_m} - B_{\mathcal{G}_m}\|_2^2$ under the constraints $\|Y_{\mathcal{G}_m}\|_2 \le \rho_m$ for all m where $B = W^{(k+1)} - \beta^{-1} Z^{(k)} + S$. The solution is given as $Y^{(k+1)} = B - \mathrm{prox}(B)$ where

$$\mathrm{prox}(B) = \left(\max(\|B_{\mathcal{G}_m}\|_2 - \rho_m, 0) \frac{B_{\mathcal{G}_m}}{\|B_{\mathcal{G}_m}\|_2} \right)_{1 \le m \le M} .$$

Hence, the first update step of DAL can be simplified as follows [13],

$$W^{(k+1)} \in \underset{W \succ 0}{\mathrm{argmin}} \; - \log\det W + \frac{\beta}{2} \left\| \mathrm{prox}\left(W + \beta^{-1} Z^{(k)} - S \right) \right\|_{\mathrm{F}}^2 .$$

This is a smooth convex optimization problem and solvable with some proper methods, e.g. via a quasi–Newton method.

4 Group SICS via DAL–ADMM

The DAL algorithm derived in the preceding section has a super–linear convergence property. This property is based on the simultaneous update of W and Y and a gradual increase of β in every steps. However, group SICS involves $\mathcal{O}(d^2)$ free parameters to be optimized and the computation of the gradient over W requires $\mathcal{O}(d^3)$ complexity. This can be too demanding even for middle sized d. Therefore, we need to reduce the number of gradient evaluations so that the entire procedure to become much more efficient. In this section, we tackle this problem by introducing an idea of Alternating Direction Method of Multipliers (ADMM) [5, 10, 14] and propose a DAL–ADMM algorithm.

In ADMM, we decouple the minimization of W and Y into sequential steps,

$$W^{(k+1)} \in \underset{W \succ 0}{\mathrm{argmin}} \, \mathcal{L}_\beta(W, Y^{(k)}, Z(k)) ,$$
$$Y^{(k+1)} \in \underset{Y}{\mathrm{argmin}} \, \mathcal{L}_\beta(W^{(k1+1)}, Y, Z^{(k)}) .$$

It means that the optimization of $\mathcal{L}_\beta(W, Y, Z^{(k)})$ over W and Y is solved only in an approximate manner. Under this relaxation, as we see later, we can construct an analytic update procedure for W which requires only one eigenvalue decomposition in every update steps. This modification has another advantage that we can use both $p_m = 2$ and ∞ while it was limited to $p_m = 2$ in DAL. On the other hand, unlike DAL, only a linear convergence is guaranteed for DAL–ADMM. However, as we sill see in numerical experiments, a reduction of the number of gradient evaluation overwhelms this drawback and results in the faster computation. In the next subsection, we detail the above two update procedures.

4.1 Solutions to Inner Optimization Problems

The inner optimization problem over W is given as follows,

$$\min_{W \succ 0} - \log \det W + \frac{\beta}{2} \left\| W - Y^{(k)} + \beta^{-1} Z^{(k)} - S \right\|_{\mathrm{F}}^{2} .$$

By setting the derivative over W to zeros, we get the optimality condition $W - (Y^{(k)} - \beta^{-1} Z^{(k)} - S) - \beta^{-1} W^{-1} = 0_{d \times d}$. Here, let $Y^{(k)} - \beta^{-1} Z^{(k)} - S = UDU^{\top}$, $D = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_d)$ be an eigenvalue decomposition. Then, we get $W^{(k+1)} = U\tilde{D}U^{\top}$, $\tilde{D} = \mathrm{diag}(\tilde{\sigma}_1, \tilde{\sigma}_2, \ldots, \tilde{\sigma}_d)$ where $\tilde{\sigma}_i = (\sigma_i + \sqrt{\sigma_i^2 + 4\beta^{-1}})/2$ [10]. Note the positive definiteness of $W^{(k+1)}$ directly follows from this result.

As we already mentioned, an optimization of Y under a fixed W is a convex constrained problem. The following is a general form for an arbitrary $p_m \in [1, \infty]$,

$$\min_{\|Y_{\mathcal{G}_m}\|_{q_m} \leq \rho_m} \frac{1}{2} \| Y_{\mathcal{G}_m} - B_{\mathcal{G}_m} \|_2^2 .$$

For $p_m = 2$, $q_m = 2$ while $q_m = 1$ for $p_m = \infty$. Solutions to both cases are available in $\mathcal{O}(|\mathcal{G}_m|)$ computational complexities [15].

4.2 Convergence

Here, we list two convergence properties of DAL–ADMM under a fixed $\beta > 0$.

1. A sequence $\{Z^{(k)}\}_{k=1}^{\infty}$ converges to the optimal parameter $Z^* = \Lambda^*$.
2. A function value $\tilde{g}(W, Y) = -\log \det W + \sum_{m=1}^{M} \delta_{\rho_m}(Y_{\mathcal{G}_m})$ converges linearly to its global minimum $\tilde{g}(W^*, Y^*)$.

These results are available as follows. We first get the optimality condition $Z^* = {W^*}^{-1}$ by setting the derivative of $\mathcal{L}_0(W, Y, Z)$ to zeros. Then, applying the general theorem for ADMM [14, 16] and recalling $W^* = {\Lambda^*}^{-1}$, the claims follow.

4.3 Implementation Details

In our implementation of DAL–ADMM, we use two *gaps* $r_{\mathrm{p}}^{(k+1)} = \|W^{(k+1)} - Y^{(k+1)} - S\|_{\mathrm{F}}$ and $r_{\mathrm{d}}^{(k+1)} = \|Y^{(k+1)} - Y^{(k)}\|_{\mathrm{F}}$ presented by Boyd et al. [14] for the termination criteria. When both of them are under a given threshold ϵ, we regard that the process has converged and stop the iteration. Here, two gaps measure how much the equality constraint in (3) and the optimality of parameters are fulfilled respectively.

The choice of an algorithm parameter β also needs some consideration in practice. Unlike DAL, we can not merely increase β in every steps since it may lead to a non–optimal solution. In the proposed algorithm, we introduce a heuristic from Boyd et al. [14]; we update β as $\beta^{(k+1)} = 2\beta^{(k)}$ for $r_{\mathrm{p}}^{(k+1)} \geq 10 r_{\mathrm{d}}^{(k+1)}$, $\beta^{(k+1)} = 0.5\beta^{(k)}$ for $r_{\mathrm{d}}^{(k+1)} \geq 10 r_{\mathrm{p}}^{(k+1)}$, and $\beta^{(k+1)} = \beta^{(k)}$ for remaining cases. This heuristic balances two gaps and makes them small simultaneously.

5 Simulations

In this section, we demonstrate the validity of DAL–ADMM through synthetic experiments. All simulations in this section have been conducted on Windows 7 (64bit), Intel Xeon W365 CPU machines with a 6GB RAM.

5.1 Data Description

In our simulations, we have generated data in the following manner. First, we give a number of gaussian variables d and its partition $d_1, d_2, \ldots, d_K$ where $\sum_{k=1}^{K} d_k = d$. For each d_k, we generate elements of a random matrix $U_k \in \mathbb{R}^{d_k \times 5d_k}$ independently from a unit normal distribution $\mathcal{N}(0,1)$. Then, generate a positive definite matrix $C_k = L_k L_k^\top$ and set the resulting precision matrix $\Lambda \in \mathbb{R}^{d \times d}$ as a block–diagonal matrix with $C_1, C_2, \ldots, C_K$ on its block–diagonal. Here, each group $\mathcal{G}_m$ corresponds to a pair of d_k and $d_{k'}$ variables with $1 \leq k, k' \leq K$ and the total number of groups is $M = K^2$. In the simulation, we considered 3 cases with $d = 20, 60, 100$. For each case, the number of partition K and a value $d_1 = d_2 = \ldots = d_K = r$ are set to $(K, r) = (2, 10), (3, 20), (4, 25)$ respectively. After a precision matrix Λ is derived, we generated $n = 5d$ independent samples from a normal distribution $\mathcal{N}(\mathbf{0}_d, \Lambda^{-1})$.

5.2 Baseline Methods

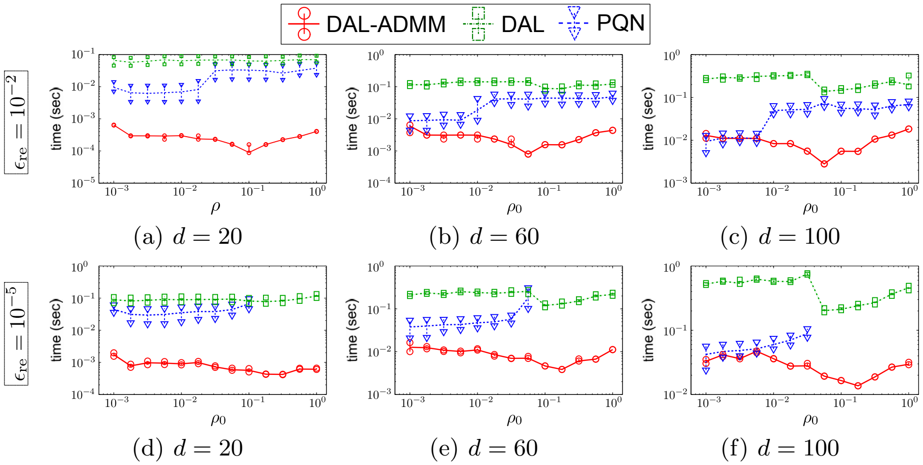
In the simulation, we adopted a PQN algorithm [15] to contrast with DAL–ADMM. We also introduced DAL to compare with DAL–ADMM aiming to observe the advantage of an ADMM relaxation. Each of DAL–ADMM, DAL and PQN are implemented using MATLAB and C. We used a DAL package [1] and implemented a DAL procedure for group SICS. We have also modified a PQN package [2] and used for our simulation. For the compatibility purpose with DAL, our experiments are conducted using a hyper–parameter $p_m = 2$ for all groups. In the simulation, we set $\rho = d\rho_0$ where ρ_0 varies in 13 different values ranging from 10^{-3} to 10^0 in a log–scale.

5.3 Results

We randomly generated datasets 1000 times for each setting and compared the running time of DAL–ADMM, DAL and PQN. The results are summarized in Fig. 1. In the figure, we plotted median times that each method achieved a relative error $(g(\Lambda^{(k)}) - g(\Lambda^*))/g(\Lambda^*)$ under tolerance parameters $\epsilon_{\text{gap}} = 10^{-2}$ and 10^{-5}. The vertical bars extend from 25% to 75% quantiles of the running time. Note PQN did not achieve a relative error under $\epsilon_{\text{re}} = 10^{-5}$ for larger ρ_0 and thus omitted from the graph.

[1] Available at http://www.ibis.t.u-tokyo.ac.jp/ryotat/dal/

[2] Available at http://www.di.ens.fr/~mschmidt/Software/PQN.html



(a) $d = 20$ (b) $d = 60$ (c) $d = 100$

(d) $d = 20$ (e) $d = 60$ (f) $d = 100$

Fig. 1. Median running time of each method until achieving a relative error under $\epsilon_{\mathrm{re}} = 10^{-2}$ and 10^{-5} with vertical bars extending from 25% to 75% quantiles

In any experimental setting, we observe that DAL–ADMM outperforms other twos. Especially, we can see the gradual decrease of the DAL–ADMM running time for larger ρ_0. We conjecture this property is what DAL original had as an efficient optimization method for sparse regularization problems, and is also inherited to DAL–ADMM. Through simulations, we observed that the inner optimization process in DAL gets a practical bottleneck and is resolved by an ADMM relaxation resulting in a dramatic improvement. A solution sequence in PQN approached to the optimal solution in a relatively small running time. However, at some point, this speed drastically decreases and the improvement of the solution seems to be bounded afterward.

6 Conclusion

In this paper, we treated a group SICS problem (2) where the state-of-the-art method for SICS (1) is no longer available. Our proposed DAL–ADMM algorithm is based on the DAL which we relaxed by introducing an ADMM approximation. In synthetic experiments, we observed that this relaxation dramatically improved the running time against naively applying DAL. A comparison of DAL–ADMM against PQN also showed favorable results that DAL–ADMM is faster and hence works well for larger ρ where PQN tends to require a longer running time.

Several future works have been indicated. The optimal choice of an algorithm parameter β remains an open problem. In our algorithm, we used a heuristic update which works practically well but does not have any theoretical guarantees. An introduction of a skipping technique in [5] would be a promising extension of DAL–ADMM to further improve its performance.

Acknowledgment. We would like to acknowledge support for this project from JSPS Grant-in-Aid for Scientific Research(B) #22300054 and Grant-in-Aid for JSPS Fellows #248850.

References

1. Dempster, A.P.: Covariance selection. Biometrics 28(1), 157–175 (1972)
2. Meinshausen, N., Bühlmann, P.: High-dimensional graphs and variable selection with the lasso. The Annals of Statistics 34(3), 1436–1462 (2006)
3. Yuan, M., Lin, Y.: Model selection and estimation in the gaussian graphical model. Biometrika 94, 19–35 (2007)
4. Banerjee, O., El Ghaoui, L., d'Aspremont, A.: Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. The Journal of Machine Learning Research 9, 485–516 (2008)
5. Scheinberg, K., Ma, S., Goldfarb, D.: Sparse inverse covariance selection via alternating linearization methods. In: Advances in Neural Information Processing Systems, vol. 23, pp. 2101–2109 (2010)
6. Duchi, J., Gould, S., Koller, D.: Projected subgradient methods for learning sparse gaussians. In: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, pp. 145–152 (2008)
7. Turlach, B., Venables, W., Wright, S.: Simultaneous variable selection. Technometrics 47(3), 349–363 (2005)
8. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B 68(1), 49–67 (2006)
9. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics 9(3), 432–441 (2008)
10. Yuan, X.: Alternating direction methods for sparse covariance selection (preprint 2009), http://www.optimization-online.org/DB_FILE/2009/09/2390.pdf
11. Scheinberg, K., Rish, I.: Learning Sparse Gaussian Markov Networks Using a Greedy Coordinate Ascent Approach. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010, Part III. LNCS, vol. 6323, pp. 196–212. Springer, Heidelberg (2010)
12. Hsieh, C., Sustik, M., Dhillon, I., Ravikumar, P.: Sparse inverse covariance matrix estimation using quadratic approximation. In: Advances in Neural Information Processing Systems, vol. 24, pp. 2330–2338 (2011)
13. Tomioka, R., Suzuki, T., Sugiyama, M.: Super-linear convergence of dual augmented lagrangian algorithm for sparsity regularized estimation. The Journal of Machine Learning Research 12, 1537–1586 (2011)
14. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning 3(1), 1–122 (2011)
15. Schmidt, M., Van Den Berg, E., Friedlander, M., Murphy, K.: Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, pp. 456–463 (2009)
16. He, B., Yuan, X.: On the $o(1/n)$ convergence rate of the douglas–rachford alternating direction method. SIAM Journal on Numerical Analysis 50, 700–709 (2012)

Multiple Outlooks Learning with Support Vector Machines

Yinglu Liu, Xu-Yao Zhang, Kaizhu Huang, Xinwen Hou, and Cheng-Lin Liu

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences,
95 Zhongguancun East Road, Beijing 100190, China
{ylliu,xyz}@nlpr.ia.ac.cn, kaser.huang@gmail.com

Abstract. Multiple Outlooks Learning (MOL) has recently received considerable attentions in machine learning. While traditional classification models often assume patterns are living in a fixed-dimensional vector space, MOL focuses on the tasks involving multiple representations or outlooks (e.g., biometrics based on face, fingerprint and iris); samples belonging to different outlooks may have varying feature dimensionalities and distributions. Current MOL methods attempted to first map each outlook heuristically to a common space, where samples from all the outlooks are assumed to share the same dimensionality and distribution after mapping. Traditional off-the-shelf classifiers can then be applied in the common space. The performance of these approaches is however often limited due to the independence of mapping functions learning and classifier learning. Different from existing approaches, in this paper, we proposed a novel MOL framework capable of learning jointly the mapping functions and the classifier in the common latent space. In particular, we coupled our novel framework with Support Vector Machines (SVM) and proposed a new model called MOL-SVM. MOL-SVM only needs to solve a sequence of standard linear SVM problems and converges rather rapidly within only a few steps. A series of experiments on the 20 newsgroups dataset demonstrated that our proposed model can consistently outperform the other competitive approaches.

Keywords: Multiple outlooks learning, multiple views learning, transfer learning, support vector machines.

1 Introduction

Traditional pattern classification models often assume that the patterns are living in a fixed-dimensional vector space. However, in practice, many learning tasks involve multiple representations, for example, biometrics based on face, fingerprint and iris. Each representation is denoted as an outlook. Samples belonging to different outlooks may have varying feature representations, e.g. different dimensionalities and distributions. Moreover, the number of samples in some particular outlook may be very small which is insufficient to achieve a high accuracy by single outlook learning. The purpose of Multiple Outlooks Learning

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 116–124, 2012.

(MOL) [5,6] is to make full use of all the information in different outlooks for boosting the classification accuracy in a particular outlook.

Multiple outlooks learning is highly related to multiple views learning (MVL) [1,7,9]. The key difference is that: in MVL each sample can be represented by all the views, which means we have the instance-correspondence across different views. Contrarily, in MOL the samples in different outlooks have no relationship except the class labels. Features from different outlooks can be extracted from different samples, and no common entities exist among different outlooks. MOL is also related to transfer learning (TL) [3,4,8], where the distributions of the training and the test data are usually different. In MOL different outlooks may also have distribution transfer. However, TL assumes the features are in the same space, but in MOL, different outlooks live in different feature spaces. In this sense, both MVL and TL can be considered as special cases of MOL.

MOL is often handled in two steps. First, a mapping function is learned for each outlook, and then all the outlooks are mapped into a common latent space, in which the mapped features from different outlooks share the same dimensionality and distribution. Therefore, the traditional classification models, such as SVM, KNN, ANN etc., can then be applied successfully in the common latent space to classify all the transformed samples from different outlooks. So far, there are two main ideas for learning the mapping functions. One is to constrain the distribution similarities of different outlooks. For example, Harel et al. [5,6] proposed an algorithm called Multiple Outlook MAPping algorithm (MOMAP), which is applied to learn an orthogonal matrix for each outlook by matching the empirical distributions of different outlooks. The other idea is to constrain the sample similarities that belong to the same class from different outlooks. In particular, Wang et al. [10] proposed a model for pursuing three goals in the common latent space: matching instances with the same labels, separating instances with different labels, and preserving topology of each given domain.

However, these models only consider to learn the mapping functions in some heuristic manners. More seriously, the learning of the mapping functions is independent with the learning of the classifier. This shortcoming limits the performance, since the mapping functions and the classifier in the common latent space are closely related to each other. To attack this problem, we proposed a novel model in this paper to learn the mapping functions and the classifier simultaneously. Our model is a combination of multiple outlooks learning and support vector machines (MOL-SVM), where the mapping function of each outlook and the SVM in the common latent space are learned jointly in an alternating optimization framework. Specifically, MOL-SVM only needs to solve a sequence of standard linear SVM problems, which can be implemented efficiently with many successful softwares such as the libSVM [2]. One appealing feature is that the involved optimization usually converges rapidly within a few epoches (typically fewer than 5 epoches in our experiment). Experiment on the 20 newsgroups dataset also demonstrated that MOL-SVM can outperform the traditional MOL models (e.g. MOMAP) and the single models (trained in each single outlook), which indicates (1) the benefits of MOL in improving the classification performance against the single task models; and (2) the advantages of joint learning of mapping functions and classifier for multiple outlooks learning.

2 Multiple Outlooks Learning with Support Vector Machines

In this section, we introduce the MOL-SVM model for multiple outlooks learning. Suppose we have K outlooks $D = \{(X^{(1)}, y^{(1)}), (X^{(2)}, y^{(2)}), \dots, (X^{(K)}, y^{(K)})\}$, where $X^{(i)} = \{X_1^{(i)}, X_2^{(i)}, \dots, X_{N_i}^{(i)}\},\ X_j^{(i)} \in R^{d_i}$ and $y^{(i)} = \{y_1^{(i)}, y_2^{(i)}, \dots, y_{N_i}^{(i)}\}$, $y_j^{(i)} \in \{1, 2, \dots, M\}$. d_i is the feature dimensionality of the i-th outlook and M is the number of classes. Here we use the superscript to denote the outlook index, and the subscript to denote the sample index. The main challenges of multiple outlooks learning (MOL) include: (1) different outlooks may have different feature dimensionalities ($d_1 \neq d_2 \neq \cdots \neq d_K$); and (2) the distributions of different outlooks may be different from each other (distribution transfer).

We first give the objective function of the traditional SVM model as:

$$\begin{aligned}\min_{W,\xi} \quad & \tfrac{1}{2}\|W\|^2 + C\textstyle\sum_{i=1}^{N} \xi_i \\ \text{s.t. } & y_i(W^T x_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \\ & \forall i = 1, \dots, N\end{aligned}$$

For multiple outlooks problems, since different samples from different outlooks are in distinct feature spaces, we cannot train a classifier using the traditional SVM. Hence, we propose to learn a mapping function $F^{(k)} \in \mathbb{R}^{d \times d_k}$ for each outlook. After the mapping, samples in different outlooks are transformed into a common latent space $\mathbb{R}^d$ where a standard SVM $\{W \in \mathbb{R}^d, b \in \mathbb{R}\}$ can be trained with all the mapped samples. The formulation of MOL-SVM is:

$$\begin{aligned}\min_{W,b,F^{(k)},\xi} \quad & \tfrac{\alpha}{2}\|W\|^2 + \textstyle\sum_{k=1}^{K} \tfrac{\beta^{(k)}}{2}\|F^{(k)}\|^2 + \sum_{k=1}^{K}\sum_{i=1}^{N_k} \xi_i^{(k)} \\ \text{s.t.} \quad & y_i^{(k)}(W^T X_{map}{}_i^{(k)} + b) \geq 1 - \xi_i^{(k)} \\ & X_{map}{}_i^{(k)} = F^{(k)} X_i^{(k)} \\ & \xi_i^{(k)} \geq 0 \\ & \forall k = 1, \dots, K \\ & \forall i = 1, \dots, N_k\end{aligned}$$

where $X_{map}{}_i^{(k)}$ refers to the mapped data from the k-th outlook. The $\|X\|^2$ denotes the L_2 norm when X is a vector and the Frobenius norm when X is a matrix. This is a quadratically constrained quadratic program (QCQP) problem. This objective yields a convex quadratic program in W, b given $\{F^{(k)}\}$, and a convex quadratic program in $\{F^{(k)}\}$ given W. Therefore we can solve the model efficiently with the alternating optimization framework.

When fixing $F^{(k)}$, the optimization of $\{W, b\}$ is a standard SVM problem with input data $\{{X_{map}}^{(k)}, k = 1, 2, \ldots, K\}$. When we fix W, the problem can be divided into K independent optimization problems of $F^{(k)}$ separately. By omitting the superscript, the general form of learning F is:

$$\begin{aligned} \min_{F,\xi} \quad & \tfrac{\beta}{2}\|F\|^2 + \textstyle\sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & y_i(W^T F X_i + b) \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \end{aligned}$$

We first reformulate the left part of the above constraints as follows.

$$\begin{aligned} W^T F X_i + b &= tr(W^T F X_i) + b \\ &= tr(F X_i W^T) + b \\ &= vec(F^T)^T vec(X_i W^T) + b \end{aligned}$$

where $vec(x)$ is the operation that spreads a matrix into a vector. Let $\tilde{F} = vec(F^T)$, $\tilde{X}_i = vec(X_i W^T)$, the formulation is equivalent to:

$$\begin{aligned} \min_{\tilde{F},\xi} \quad & \tfrac{\beta}{2}\|\tilde{F}\|^2 + \textstyle\sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & y_i(\tilde{F}^T \tilde{X}_i + b) \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \end{aligned}$$

We can see this problem is also a standard SVM problem with input data $\tilde{X}_i$.

The complete procedure for MOL-SVM is to solve two standard linear SVM problems repeatedly until convergence. After that we can get the classifier $\{W, b\}$ and the mapping functions $F^{(k)}$ for each outlook. We summarize the MOL-SVM model in Algorithm 1.

3 Experiment

In this section, we conducted a series of experiments on the 20 Newsgroups dataset and compared our proposed MOL-SVM with several other methods, including the single outlook learning and the combined outlooks learning with different mapping methods.

3.1 Dataset Description

The 20 Newsgroups dataset[1] is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. We consider the main topics as classes, and the subtopics as outlooks. For example,

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/

Algorithm 1. MOL-SVM

```
Input: $D = \{(X^{(1)}, y^{(1)})\},\ (X^{(2)}, y^{(2)}),\ \ldots,\ (X^{(K)}, y^{(K)}),\ \epsilon$
  Initialize $F^{(k)},\ k = 1, 2, \ldots, K$
  repeat
    Calculate mapped data for SVM learning
      $X_{map} = [F^{(1)}X^{(1)}\ F^{(2)}X^{(2)}\ \ldots\ F^{(K)}X^{(K)}]$
    SVM learning $W, b$
      $\min_{W,b}\ \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{N} \max(0, 1 - (W^T X_{map_i} + b))$
    Calculate transformed data for mapping functions learning
      $\tilde{X}^{(k)} = vec(X^{(k)}W^T)\ \ \forall k = 1, 2, \ldots, K$
    Mapping functions learning $\tilde{F}^{(k)},\ \ k = 1, 2, \ldots, K$
      $\min_{\tilde{F}^{(k)},b}\ \frac{1}{2}\|\tilde{F}^{(K)}\|^2 + C\sum_{i=1}^{N_k} \max(0, 1 - (\tilde{F}^{(k)T}\tilde{X}_i^{(k)} + b))$
    Revive the mapping functions
      $\left\{vec^{-1}(\tilde{F}^{(k)})\right\}^T \rightarrow F^{(k)}$
  until $\|\Delta W\|^2 \leq \epsilon$
Output: $F^{(k)},\ k = 1, 2, \ldots, K$ and $\{W, b\}$
```

rec.sport.baseball and rec.sport.hockey belong to the same class (sport), while they are from two outlooks. In our experiment, we select rec.sport.baseball and talk.politics.misc as outlook 1, and rec.sport.hockey and talk.politics.mideast as outlook 2.[2]

3.2 Experimental Setting

Our purpose is to classify two classes of outlook 1 (rec.sport.baseball and talk.politics.misc). Moreover, we assume that the labeled samples of outlook 1 are limited (by taking a few samples of outlook 1 for training), and the labeled data of outlook 2 are sufficient (by using all the samples for training). Hence, the objective of multiple outlooks learning (MOL) is to utilize the outlook 2's information to improve the outlook 1's classification accuracy. We compared our method with several other algorithms in two cases: **Case 1** when the data from two outlooks have the same dimensionality, **Case 2** when they have different dimensionalities. The compared methods are listed as follows:

- **Outlook 1(original)**: we trained the classifier just using the data from outlook 1.
- **Combined outlooks**: we mapped the data from outlook 1 and outlook 2 to a common space, and used the combined mapped data in the common space to train the classifier. We evaluated three different mapping functions.
  - **original**: we used the original combined data without mapping.
  - **MOMAP**: The Multiple Outlook MAPping (MOMAP) algorithm proposed by [6] is aimed to learn the mapping functions by matching the

[2] Experiments on another group of data (Outlook 1: comp.sys.ibm.pc.hardware & sci.med; Outlook 2: comp.sys.mac.hardware & sci.space) also revealed the best performance of our model. Due to the space limitation, we omitted the results in this paper.

mean and the principle directions of two outlooks. The goal is to map the outlook with more information (outlook 2) to the outlook with little information (outlook 1). We learned a mapping function by MOMAP, mapped the data from outlook 2 to outlook 1, and then trained the classifier with the combined mapped data in the common space (outlook 1).

- **MOL-SVM**: the detailed procedure of the proposed MOL-SVM model is shown in Algorithm 1. Note that MOL-SVM is to map the outlook with little information to the outlook with more information (from outlook 1 to outlook 2), which is opposite to MOMAP.

In order to compare the performance of the mapping functions more obviously, we also tested the methods by just using the mapped data of outlook 2 for training, and testing on the data of outlook 1.

- **Outlook 2**: we trained the classifier just using the mapped data of outlook 2.
    - **original**: we used the original data of outlook 2 without mapping.
    - **MOMAP**: we learned a mapping function by MOMAP, mapped the data of outlook 2 to the common space (outlook 1), and trained the classifier with the mapped data of outlook 2.
    - **MOL-SVM**: we learn a mapping function by MOL-SVM, train the classifier with the original data of outlook 2, and map the test data of outlook 1 to the common space (outlook 2) for testing.

In Case 1, we evaluated all the mentioned above methods, while in Case 2, we just compared Outlook 1 (original), Combined outlooks (MOL-SVM), Outlook 2 (original, MOMAP, MOL-SVM). This is because when the feature dimensionalities from two outlooks are different, the traditional combined method (combine outlook 1 and outlook 2 without mapping) will not be applicable. Note that, In [6], MOMAP merely used the mapped data of outook 2 (without outlook 1) for training, but we find that the combined outlooks will improve the performance in Case 1 due to the same dimensionality while it does not work well in Case 2. Therefore, we evaluated this method in Case 1 but omitted it in Case 2.

We now report the experimental setup. The original dimensionality of features is 61,188, which is difficult to process. For simplicity, we reduced the dimensionality by PCA after normalizing each dimension with zero mean and unit variance. In Case 1, we reduced the dimensionality to 30 for both outlook 1 and outlook 2. In Case 2, we reduced the final dimensionality of outlook 1 to 20 and outlook 2 to 10. We used all the training data from outlook 2 (the total number is 1,162) and different numbers of training data from outlook 1 (the total number is 1,058), and calculated the test error rate (the number of test data from outlook 1 is 707). While it is difficult to apply cross validation when the number of training data from outlook 1 is too small, we sampled 20 samples from outlook 1's test dataset as the validation dataset (the remaining as the test dataset) and selected the parameters on it.

Table 1. Test set error rate comparison (Case 1)

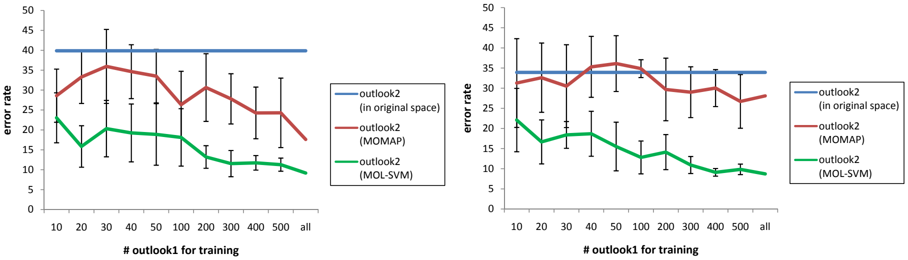
# training	outlook1	combined outlooks	combined outlooks	combined outlooks
(outlook1)	(original)	(original)	(MOMAP)	(MOL-SVM)
10	22.94 (± 5.15)	38.25 (± 1.09)	27.88 (± 5.46)	20.31 (± 5.53)
20	17.25 (± 2.20)	35.36 (± 1.51)	25.05 (± 7.19)	14.45 (± 3.86)
30	17.03 (± 4.55)	35.02 (± 1.43)	29.96 (± 7.23)	16.39 (± 3.97)
40	16.16 (± 3.55)	33.72 (± 1.89)	24.99 (± 7.63)	15.02 (± 4.31)
50	14.91 (± 4.33)	30.79 (± 2.23)	23.12 (± 5.90)	13.19 (± 3.86)
100	13.10 (± 3.67)	27.07 (± 1.11)	20.23 (± 7.16)	12.80 (± 2.49)
200	12.79 (± 2.91)	23.12 (± 1.88)	23.77 (± 6.12)	11.09 (± 1.48)
300	10.99 (± 1.43)	20.20 (± 1.47)	17.99 (± 6.48)	9.84 (± 1.76)
400	12.20 (± 3.25)	19.90 (± 2.15)	19.46 (± 5.46)	9.59 (± 1.29)
500	10.25 (± 1.52)	18.18 (± 1.01)	15.60 (± 3.69)	9.27 (± 0.68)
all	8.59 (± 0.00)	14.99 (± 0.00)	11.5 (± 0.00)	8.44 (± 0.00)

Table 2. Test set error rate comparison (Case 2)

# training	outlook1	combined outlooks
(outlook1)	(original)	(MOL-SVM)
10	22.21 (± 7.91)	20.35 (± 7.58)
20	16.64 (± 3.29)	16.68 (± 4.60)
30	17.15 (± 3.10)	16.17 (± 2.88)
40	14.06 (± 2.72)	13.81 (± 3.15)
50	12.96 (± 3.63)	12.49 (± 3.01)
100	11.28 (± 2.44)	10.86 (± 2.50)
200	11.99 (± 2.53)	11.11 (± 2.45)
300	10.57 (± 1.89)	9.52 (± 1.43)
400	9.97 (± 1.86)	8.82 (± 0.55)
500	10.38 (± 1.52)	8.41 (± 0.62)
all	8.59 (± 0.00)	8.59 (± 0.00)

3.3 Experimental Result

We evaluated several algorithms under different numbers of training samples of outlook 1. For each number we sampled randomly from outlook 1 for 10 times, and reported the average error rates on the test set. Table 1 presents the performance in Case 1. Table 2 shows the results in Case 2. We can see our method presents the best performance on both cases. Figure 1 shows the results when we just used the mapped data of outlook 2 for training. The x-axis presents the sampled number of training data from outlook 1, and the y-axis refers to the error rate. We can see our method is significantly better than MOMAP and the original combined method. In this experiment, the performance of combined outlooks by MOMAP is not as good as single outlook. Since it is difficult to evaluate the distribution when the sample number is small, and also difficult to get the proper mapping functions which can well match the two spaces. However, our method does not minimize the distance among outlooks or among samples, but focuses on minimizing the classifier output of the same class among different outlooks. This is much easier to be satisfied and hence can lead to more robust performance. The optimization usually converged rapidly within 5 epoches.



(a) Case 1

(b) Case 2

Fig. 1. Test set error rate with different mapping functions

4 Conclusion

In this paper, we proposed a novel method to deal with the multiple outlooks problem, where the features from different outlooks have no common entities and different outlooks have different dimensionalities and distinct distributions. Compared to other multiple outlooks methods, we learned the mapping functions jointly with the classifier, which proves more accurate and robust. Moreover, our model is to constrain the output similarities (the label space) among different outlooks, while the traditional method is to constrain the input similarities among different outlooks (e.g. matching the empirical distributions of feature space). Therefore our model could be more readily to be satisfied when the features are from quite different spaces. A series of experiments on real data demonstrated the effectiveness of our model. In the future, we will extend our method to the kernel space for solving the non-linear problems.

Acknowledgements. This work has been supported in part by the National Basic Research Program of China (973 Program) Grant 2012CB316301, the National Natural Science Foundation of China (NSFC) Grants 61075052 and 60825301, and Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation.

References

1. Amini, M., Usunier, N., Goutte, C.: Learning from Multiple Partially Observed Views-an Application to Multilingual Text Categorization. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 28–36 (2009)
2. Chang, C., Lin, C.L.: a library for support vector machines (2001) Software, `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
3. Dai, W., Yang, Q., Xue, G., Yu, Y.: Boosting for Transfer Learning. In: Proceedings of International Conference on Machine Learning, pp. 193–200 (2007)
4. Dai, W., Chen, Y., Xue, G., Yang, Q., Yu, Y.: Translated learning: transfer learning across different feature spaces. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 353–360 (2008)

5. Harel, M., Mannor, S.: Learning from Multiple Outlooks (2011), http://arxiv.org/abs/1005.0027v1
6. Harel, M., Mannor, S.: Learning from Multiple Outlooks. In: Proceedings of the International Conference on Machine Learning (2011)
7. Hou, C., Zhang, C., Wu, Y., Nie, F.: Multiple view semi-supervised dimensionality reduction. Pattern Recognition 43(3), 720–730 (2010)
8. Pan, S.J., Yang, Q.: A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering 22(10), 1345–1359 (2010)
9. Rüping, S., Scheffer, T.: Learning with Multiple Views. In: Proceeding of the International Conference on Machine Learning Workshop on Learning with Multiple Views (2005)
10. Wang, C., Mahadevan, S.: Heterogeneous Domain Adaptation Using Manifold Alignment. In: Proceeding of the 22nd International Joint Conference on Artificial Intelligence, pp. 1541–1546 (2011)

Multi-Task Learning Using Shared and Task Specific Information

P.K. Srijith and Shirish Shevade*

Computer Science and Automation
Indian Institute of Science, Bangalore
{srijith,shirish}@csa.iisc.ernet.in

Abstract. Multi-task learning solves multiple related learning problems simultaneously by sharing some common structure for improved generalization performance of each task. We propose a novel approach to multi-task learning which captures task similarity through a shared basis vector set. The variability across tasks is captured through task specific basis vector set. We use sparse support vector machine (SVM) algorithm to select the basis vector sets for the tasks. The approach results in a sparse model where the prediction is done using very few examples. The effectiveness of our approach is demonstrated through experiments on synthetic and real multi-task datasets.

Keywords: Multi-task learning, Support vector machines, Kernel methods, Sparse models.

1 Introduction

Multi-task learning (MTL) is used in situations where one has to solve several related learning problems. Multi-task learning models each learning problem as a separate task but instead of learning the tasks independently, learns them together [1]. It is extremely effective when each learning problem is associated with a limited dataset. It enables a task to be learnt using the data from multiple related tasks. This results in a better predictive performance of the individual tasks. It has been shown that multi-task learning performs better than learning tasks independently [2,3,4]. Multi-task learning methods are successfully applied to applications like user preference modeling [5] and conjoint analysis [6].

Multi-task learning (MTL) has recently created a lot of interest in the machine learning community. Many approaches have been proposed to effectively learn multiple related tasks. Task similarity could be captured by restricting different task functions to be close to each other in some distance measure [4]. Bayesian approaches [5] capture task similarity by sharing a common prior among different tasks. Other approaches capture task similarity by sharing a common internal representation [2,3] across all the tasks. Most of the kernel based multi-task learning approaches [4] use all the training examples to make a prediction, resulting in higher computational and storage requirements. Also, they try to capture only task similarity and not task variability.

* The author gratefully acknowledges the support for this work from Infosys Ltd., India.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 125–132, 2012.

We propose a novel multi-task learning model in which task similarity is captured by sharing a common set of basis vectors across all tasks. In addition, it has task specific basis vectors which capture the variability across different tasks. It uses both the common set and the task specific set to make predictions for the test data. The use of both the common set and the task specific set helps it to capture the relatedness between tasks more effectively. The basis vector sets are learnt by extending the sparse SVM algorithm [7] for single task learning to the multi-task scenario. It results in a sparse model which requires very few training examples to make a prediction. This enables it to make predictions much faster and is extremely useful when dataset size is large. Experimental results on synthetic and real datasets show the usefulness of our approach.

We discuss some related work in section 2. Section 3 discusses the proposed sparse multi-task learning approach in detail. Section 4 presents the experimental results of running the proposed approach on synthetic and real multi-task datasets. Finally we conclude in section 5.

2 Related Work

We consider multi-task learning problems with T tasks. Each task t is associated with a dataset D_t with m_t examples, *i.e.* $D_t = \{x_{ti}, y_{ti}\}_{i=1}^{m_t}$. Let $n = \sum_{t=1}^{T} m_t$ be the total number of examples from all the tasks. Each task specific dataset D_t comes from the same input and output space $X \times Y$ where $X \subset \mathcal{R}^d$ and $Y = \{+1, -1\}$ for classification or $Y \subset \mathcal{R}$ for regression. We assume that task specific datasets are associated with a different but related sampling distributions P_t. The goal is to learn T functions $f_1, f_2, \ldots, f_T$ for T tasks such that each task specific function f_t gives good generalization performance.

Regularized multi-task learning [8] captures similarity among tasks by assuming all the task specific functions to be close to each other. The parameters of task specific functions are learnt using the modified SVM framework. The approach uses almost the entire training examples to make a prediction. The proposed sparse multi-task learning approach differs from it in using very few training examples to make a prediction. Radial basis function network for multi-task learning [9] captures similarity by sharing the basis functions across all the tasks. The proposed approach differs from it in having a task specific basis function set in addition to the shared set. It enables the proposed approach to more effectively capture the task relations.

3 Sparse Multi-Task Learning

Sparse multi-task learning approach captures the similarity between tasks by restricting the task specific functions to share some common structure. It models this by assuming the predictive function for a task to have a common part shared by all the tasks and a task specific part particular to the task. It represents the predictive function for a task t as $f_t(x) = w_c.\phi(x) + w_t.\phi(x)$, where ϕ is a feature map which maps the examples from input space X to some reproducing kernel

Hilbert space H with an associated kernel function K. The common part w_c captures the similarity among the tasks, while the task specific part w_t captures the variability across different tasks. We assume that the common part w_c takes the basis function expansion form $w_c = \sum_{c=1}^{N} \alpha_c \phi(x_c)$ where $\phi(x_c)$ is the basis associated with the example x_c in the common basis vector set C of size N and α is the parameter associated with the common set C. Examples in the common basis vector set C could belong to any task. Similarly the task specific part w_t for a task t is represented as $w_t = \sum_{j=1}^{M_t} \beta_{tj} \phi(x_{tj})$, where $\phi(x_{tj})$ is the basis associated with the example x_{tj} in the task specific basis vector set J_t of size M_t and β_t is the parameter associated with the task specific set J_t. Examples in the task specific basis vector set J_t belongs only to task t. Using the basis function expansion form and kernels to represent the inner product between basis functions, the predictive function f_t for a task t could be written as

$$f_t(x) = K_{xC}.\alpha^\top + K_{xJ_t}.\beta_t^\top \tag{1}$$

where $K_{xC} = [K(x, x_1), \ldots, K(x, x_N)]$, $\alpha = [\alpha_1, \ldots, \alpha_N]$, $K_{xJ_t} = [K(x, x_{t1}), \ldots, K(x, x_{tM_t})]$, $\beta_t = [\beta_{t1}, \ldots, \beta_{tM_t}]$ and $K(x_i, x_j) = \phi(x_i).\phi(x_j)$.

The selection of basis vector sets and the estimation of parameters for multi-task classification problems are done by minimizing the objective function

$$\underset{\alpha,\beta_1,\ldots,\beta_T,C,J_1,\ldots,J_T}{\arg\min} \; \frac{\gamma}{2}\alpha K_{CC}\alpha^\top + \frac{\lambda}{2}\sum_{t=1}^{T}\beta_t K_{J_tJ_t}\beta_t^\top + \frac{1}{2}\sum_{t=1}^{T}\sum_{i\in I_t}(1 - y_{ti}o_{ti})^2 \tag{2}$$

where γ and λ are regularization parameters, K_{CC} is the kernel matrix formed from examples in the common set C, $K_{J_tJ_t}$ is the kernel matrix formed from examples in the task set J_t, $o_{ti} = K_{iC}\alpha^\top + K_{iJ_t}\beta_t^\top$ is the output of the i^{th} example belonging to task t, and $I_t = \{i : 1 - y_{ti}o_{ti} > 0\}$.

The regularization parameter controls the common and task specific basis vector sizes. A low value of γ relative to λ selects more common basis vectors than task specific basis vectors. This is ideal for the situations in which tasks are similar to each other. In the limit when $\frac{\gamma}{\lambda}$ tends to 0, this is equivalent to combined task learning in which a single classifier is learned by pooling together data from all the tasks. But in situations where tasks are dissimilar it is appropriate to choose a low value for λ relative to γ resulting in the selection of more task specific basis vectors than common basis vectors. In the limit when $\frac{\lambda}{\gamma}$ tends to 0,this is equivalent to single task learning in which tasks are learnt independently using their respective datasets.

The basis vector sets and the parameters are obtained by extending the sparse SVM [7] approach for single task learning to the multi-task learning scenario. We select the common basis vector set and task basis vector sets incrementally. After each basis selection we re-estimate the common and task parameters. Section 3.1 discusses the procedure to estimate the parameters assuming we have selected common basis vector set C and task basis vector sets J_t's. Section 3.2 discusses the procedure to select the common and task basis vector sets.

3.1 Parameter Estimation

Sparse MTL approach needs to estimate $T+1$ parameters, one common parameter α and T task parameters β_t. We use an alternative optimization approach to estimate the parameters. The approach minimizes the objective function (2) with respect to one of the parameter keeping others fixed. This is repeated for each parameter and the entire procedure is continued until the relative decrease in objective function value becomes small. Algorithm (1) describes the parameter estimation procedure in detail.

```
Algorithm 1
Procedure Parameter Estimation
Input: $C, J_1, \ldots, J_T$
Output: $\alpha$, $\beta_1, \ldots, \beta_T$
1.  Set k = 0. Choose suitable starting vectors $\alpha^{(0)}$, and ${\beta_t}^{(0)}$ for each task $t$.
    repeat
2.      For the current values of task parameters(${\beta_t}^{(k)}$), obtain $\alpha^{(k+1)}$ by minimizing the objective function (2) with respect to $\alpha$ using Newton method with line search.
3.      for each task $t$
4.          For the current value of common parameter($\alpha^{(k+1)}$), obtain ${\beta_t}^{(k+1)}$ by minimizing the objective function (2) with respect to $\beta_t$ using Newton method with line search.
5.      k ←k+1
6.  until relative decrease in objective function value (2) is small.
```

The parameter estimation uses the Newton method and it requires the calculation of the gradients and the generalized Hessians of the objective function (2). The gradients and the generalized Hessians of the objective function (2) with respect to the parameters α and β_t's are

$$\begin{aligned} g_\alpha &= \gamma K_{CC}\alpha^\top - \textstyle\sum_{t=1}^T K_{CI_t}[y_{I_t} - o_{I_t}] \quad & P_\alpha &= \gamma K_{CC} + \sum_{t=1}^T K_{CI_t}K_{I_tC} \\ g_{\beta_t} &= \lambda K_{J_tJ_t}\beta_t^\top - K_{J_tI_t}[y_{I_t} - o_{I_t}] \quad & P_{\beta_t} &= \lambda K_{J_tJ_t} + K_{J_tI_t}K_{I_tJ_t}\forall t \end{aligned} \tag{3}$$

Here g and P denote the gradient and the generalized Hessian respectively with respect to the parameters denoted by the subscripts, y_{I_t} is the column vector of labels from task t indexed by I_t, and o_{I_t} is the column vector of outputs from task t indexed by I_t.

3.2 Basis Vector Selection

The basis vectors are selected greedily in an incremental mode. The selection involves adding basis vectors to the common set and to T task specific sets. The basis vectors for the common set are obtained from examples from all the tasks while basis vectors for the task specific set are obtained from the task specific examples. Common basis vectors are selected first and task specific basis vectors are selected later. The basis vectors are selected until the relative decrease in the objective function value becomes small. Alternatively, one can predefine the number of basis vectors to be selected. The basis vector selection procedure results in a sparse model with very few elements in the common set and task specific sets. Algorithm 2 describes the basis vector selection procedure in detail.

Algorithm 2

Procedure *Basis Vector Selection*

```
repeat
    Select a basis vector from the complete training data.
    Add the selected basis vector to the common set C.
    Perform parameter estimation using Algorithm 1.
until relative decrease in objective function value is small
for each task t
    repeat
        Select a basis vector from the training dataset $D_t$.
        Add the selected basis vector to the task specific set $J_t$.
        Perform parameter estimation using Algorithm 1.
    until relative decrease in objective function value is small
```

During basis vector addition, a basis vector is selected from the training set which results in maximum decrease in objective function value on addition of it to the basis set. Selecting a basis vector from the entire training set is time consuming. Hence the basis vector is selected from a candidate set of size κ containing κ examples selected randomly from the training set. During basis selection, the objective function is optimized only with respect to the parameter corresponding to the newly added basis vector. In this case the objective function is a simple quadratic function in the variable of optimization. Therefore the optimization variable and the decrease in the objective function value can be calculated analytically. Let the newly added basis vector to the common set be c and the parameter corresponding to it be α_c. Then α_c and the reduction in the objective value is given by $-g_{\alpha_c}/P_{\alpha_c}$ and $g_{\alpha_c}^2/P_{\alpha_c}$ respectively, where

$$g_{\alpha_c} = \gamma K_{cC}\alpha - \sum_{t=1}^{T}\sum_{i\in I_t} K_{cI_t}(y_{I_t} - o_{I_t}) \qquad P_{\alpha_c} = \gamma K_{cc} + \sum_{t=1}^{T} K_{cI_t}K_{I_t c} \tag{4}$$

We could obtain similar expressions for task specific basis vector selection also. After each basis vector addition we obtain new values of the parameters by following the parameter estimation procedure described in section 3.1.

Newton method and basis vector selection require modifications in the generalized Hessian due to changes in the sets I_t, C and J_t. It could be done cheaply by maintaining a Cholesky decomposition of the generalized Hessian [7] and using efficient rank one updates [10]. Let the number of training examples in each task specific dataset be m and total number of examples be n ($n = Tm$). Let current number of elements in the common basis vector set be N and task specific basis vector set be M. On addition of a new common basis vector the cost incurred for computing new elements of the generalized Hessian and maintaining its Cholesky decomposition are $\mathcal{O}(TmN)$ and $\mathcal{O}(N^2)$ respectively. Assuming $N \ll n$ the cost of a single common basis vector addition is $\mathcal{O}(\kappa nN)$. Setting maximum number of common basis vectors to N_{max}, common basis vector set selection takes $\mathcal{O}(\kappa nN_{max}^2)$ time. Similarly the addition of a single task specific basis vector takes $\mathcal{O}(\kappa mM)$ time(assuming $M < m$). Setting maximum number of task specific basis vectors to M_{max}, task specific basis vector selection takes $\mathcal{O}(\kappa mM_{max}^2)$ time. For all tasks it takes $\mathcal{O}(\kappa TmM_{max}^2) = \mathcal{O}(\kappa nM_{max}^2)$ time.Hence the time complexity of the proposed approach is $\mathcal{O}(\kappa nN_{max}^2) + \mathcal{O}(\kappa nM_{max}^2)$.

Multi-task regression problems are solved in a similar way to the classification problems. The difference comes in the objective function which uses a least squares loss function instead of the squared hinge loss function used in the multi-task classification problem (5). Multi-task regression problem minimizes the following objective function.

$$\arg\min_{\alpha,\beta_1,\ldots,\beta_t,C,J_1,\ldots,J_T} \frac{\gamma}{2}\alpha K_{CC}\alpha^\top + \frac{\lambda}{2}\sum_{t=1}^{T}\beta_t K_{J_tJ_t}\beta_t^\top + \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{m_t}(y_{ti} - o_{ti})^2 \quad (5)$$

The calculation of gradients and generalized Hessians for the objective function (5) is similar to (3) except that I_t contains the entire training examples from task t.

4 Experiments

We conduct experiments for both multi-task classification and regression problems. Classification experiments are conducted on a synthetic dataset. Regression experiments are done on a real dataset. We compare the proposed sparse multi-task learning model (sparseMTL) against regularized multi-task learning (regMTL) [4], combined task learning (sparseCTL) and single task learning (sparseSTL). SparseCTL is learnt by pooling together data from all the tasks and then learning a single model on the combined dataset using sparse SVM [7]. The results reported for sparseCTL use the same number of basis vectors as sparseMTL. In sparseSTL each task is learnt independently using sparse SVM on the dataset corresponding to that task. The results reported for sparseSTL use the entire training dataset corresponding to the task as the basis vector set. All the experiments use the Gaussian kernel, $K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma_d^2}\right)$.

4.1 Multi-task Classification

Multi-task classification experiments are done on a synthetic dataset. The synthetic data models the preferences of individuals while choosing products. The dataset is simulated as described in [4]. The synthetic data consists of 30 tasks. Every task is associated with 96 training examples and 96 test examples. In total

Table 1. Mean misclassification error and mean number of basis vectors (given in brackets) for regMTL, sparseMTL, sparseSTL and sparseCTL on the synthetic data. Bolded column indicates the best result. We also report mean number of common basis vectors and task basis vectors obtained for sparseMTL on the synthetic data.

Similarity	RegMTL	SparseMTL	SparseCTL	SparseSTL
High	8.16%(964)	**7.22%**(60)	15.59%(60)	10.49%(96)
Low	9.79%(1330)	**9.40%**(70)	29.06%(70)	10.44%(96)

Similarity	Common	Task
High	18	42
Low	12	58

there are 2880 training and 2880 test examples. We consider two kinds of synthetic data, one in which tasks are less similar and the other in which tasks are more similar. For both the cases we consider synthetic datasets with low noise. Table 1 shows the mean misclassification error of different approaches over 5 independent runs on the dataset. It also reports the mean number of common and task basis vectors obtained for sparseMTL. Each approach is run for different hyper-parameter value settings and the best among those is reported.

We could observe from Table 1 that sparseMTL gives the best result for both the high similarity and the low similarity dataset. In addition the sparseMTL approach provides an advantage in terms of number of basis vectors needed for prediction. SparseMTL requires an order of magnitude less number of basis vectors than regMTL and performs better than regMTL. In the low similarity case sparseMTL is found to select relatively less number of common basis vectors and more number of task specific basis vectors in order to capture the dissimilarity among tasks . In the high similarity case, it is found to select relatively more number of common basis vectors and less number of task specific basis vectors.

4.2 Multi-task Regression

Multi-task regression experiments are done on school dataset[4]. The dataset consists of examination records of 15362 students over 139 schools. Each student record has 27 dimensions and the number of student records associated with each school varies from 20-150. The goal is to predict exam scores of students from each school. We used 75% of the examples from each task as the training set and the remaining 25% as the test set. In total the training data contains 11472 examples and the test data contains 3890 examples. The performance metric used is the explained variance[4] which is defined as $1 - \frac{\text{sum squared error}}{\text{total variance}}$. Table 2 reports the mean explained variance and the mean number of basis vectors required for regMTL, sparseMTL, sparseSTL, and sparseCTL over 10 independent runs on the school dataset. It also reports the mean number of common and task basis vectors selected by sparseMTL.

We could observe from Table 2 that sparseMTL performance is marginally better than regMTL. More importantly it could achieve this performance with very less number of basis vectors. SparseMTL is found to select more number of common basis vectors than task basis vectors capturing the high similarity among the tasks in the school dataset.

Table 2. Mean explained variance and mean number of basis vectors (given in brackets) for regMTL, sparseMTL, sparseSTL and sparseCTL on the school data. Bolded column indicates the best result among the different approaches. In sparseSTL every task uses the entire training data corresponding to it as the basis vector set and its size varies across tasks. We also report the mean number of common and task basis vectors selected by sparseMTL on the school dataset.

RegMTL	SparseMTL	SparseCTL	SparseSTL
0.3275(11330)	**0.3282**(75)	0.2833(75)	0.2710

Total	Common	Task
75	65	10

5 Conclusion

We proposed a novel approach to multi-task learning which captures the task relationships through common and task specific basis vector sets. We also developed an approach to select the basis vector sets. It resulted in a sparse multi-task model which uses very few training examples for predictions. The sparse model can handle very large datasets and makes predictions faster. Experimental results showed that the proposed approach was able to achieve the generalization performance close to that achieved by other multi-task approaches with very few number of basis vectors. The proposed approach, however, is not directly applicable to multi-label classification problems since the tasks share the dataset.

References

1. Caruana, R.: Multitask Learning. Machine Learning 28(1), 41–75 (1997)
2. Ando, R.K., Zhang, T.: A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. J. Mach. Learn. Res. 6, 1817–1853 (2005)
3. Bakker, B., Heskes, T.: Task Clustering and Gating for Bayesian Multitask Learning. J. Mach. Learn. Res. 4, 83–99 (2003)
4. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning Multiple Tasks with Kernel Methods. J. Mach. Learn. Res. 6, 615–637 (2005)
5. Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multi-task Learning for Classification with Dirichlet Process Priors. J. Mach. Learn. Res. 8, 35–63 (2007)
6. Argyriou, A., Evgeniou, T., Pontil, M.: Convex Multi-task Feature Learning. Machine Learning 73(3), 243–272 (2008)
7. Keerthi, S.S., Chapelle, O., DeCoste, D.: Building Support Vector Machines with Reduced Classifier Complexity. J. Mach. Learn. Res. 7, 1493–1515 (2006)
8. Evgeniou, T., Pontil, M.: Regularized Multi-task Learning. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 109–117. ACM (2004)
9. Liao, X., Carin, L.: Radial Basis Function Network for Multi-task Learning. In: Advances in Neural Information Processing Systems 18, pp. 795–802. MIT Press (2006)
10. Seeger, M.: Low Rank Updates for the Cholesky Decomposition. Technical report, University of California at Berkeley (2004)

Learning Attentive Fusion of Multiple Bayesian Network Classifiers

Sepehr Eghbali[1], Majid Nili Ahmadabadi[1,2], Babak Nadjar Araabi[1,2], and Maryam Mirian[1]

[1] Cognitive Robotics Lab, Control and Intelligent Processing Center of Excellence, School of ECE., College of Eng., Univ. of Tehran

[2] School of Cognitive Sciences, Institute for Research in Fundamental Sciences, IPM, Iran

{s.eghbali,mnili,araabi,mmirian}@ut.ac.ir

Abstract. Using Bayesian networks (BNs) for classification tasks has received significant attention as BNs can encode and represent domain-experts' knowledge as well as data in their structures and conditional probability tables. While structure learning and constructing the structure by hand according to an ensemble of domain-expert opinions are two common approaches to make a BN structure, finding an optimal structure to attain a high correct classification rate -especially for high dimensional problems- is still a challenging task. In this paper we propose a framework - called Local Bayesian Network Experts Fusion (LoBNEF) - in that, instead of making a single network, multiple Bayesian Network Classifiers (BNCs) are built and their outputs are attentively fused. The attentive fusion process is learned interactively using a Bayesian reinforcement learning method. We demonstrate that learning different BNCs in the first step and then fusing their decisions in an attentive and sequential manner is an efficient and robust method in terms of correct classification rate.

Keywords: Bayesian Network, Decision Fusion, Reinforcement Learning.

1 Introduction

Applying Bayesian network (BN) for classification task has received significant attention recently. While generative BNs aim at finding a description of whole data, discriminative BNs -a.k.a.Bayesian network classifiers (BNC)- focus on finding models which accurately discriminate between different class labels given attributes (features). In both contexts, finding the optimal BN is a demanding task especially when the number of features grows. Since a BN is defined by a structure and a set of parameters, they must be specified either by incorporating learning methods or using domain-expert knowledge.

Given the BNC structure, first attempts for discriminative parameter learning were made by [1]. They proposed a method, *Extended Logistic Regression* (ELR), which tries to minimize conditional log-likelihood by using a gradient descent

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 133–140, 2012.

approach. Although ELR is a computationally demanding method, it leads to more accurate classifiers in comparison to generative parameter learning methods such as maximum likelihood. Later, Su et al. [2] suggested *Discriminative Frequency Estimate* (DFE). DEF significantly involves less computations but results in classifiers which are nearly as accurate as those built by ELR method.

Finding a near optimal structure is another challenge in the context of BNCs. While some methods use a fixed structure, such as naive Bayes, others try to learn the structure from data. Friedman et al. [3] proposed *Tree Augmented Naive Bayes* (TAN) algorithm. TAN algorithm limits the number of parents an attribute variable can have and find a globally optimal structure subject to this constraint. Using hill-climbing with different scoring functions has also been used for discriminative structure learning. Grossman and Domingos [4] used hill-climbing in order to find a structure that maximizes conditional log-likelihood (CLL) score. More recently, Carvalho et al. [5] have managed to estimate CLL score via a factorization, called *Factorized Conditional Log-Likelihood* (FCLL), which exhibits promising characteristics such as decomposability.

As mentioned, another approach to develop the structure of a BNC is to use domain-experts' knowledge. The main challenge in this approach is aggregation of experts' opinions as different experts may have dissimilar opinions regarding the structure. In other words, it is very likely that each expert's speciality is not complete over the high dimensional problem space. It means that not only is very probable that each expert's perception over the problem domain is partial, it also can mislead the final decision making when they try to respond to questions they have never experienced before. Disparity among experts can be rooted in having different experiences, having access to different feature spaces, and dissimilarity in paying attention to feature set in addition to having different beliefs.

Moreover, in many applications, like medical ones, data collection is done in different locations. In such cases diversity in the recorded features across the recording sites is usual. It is equivalent to having many missing entries. Constructing a single BNC either by learning or by hand- using such a dataset is a challenge. Therefore, the question here is if making a single BNC -using data or experts' knowledge- is better than making different BNCs and then learning to fuse their outputs.

Using different BNCs for classification is not a new approach; see [3], [6] and [7] as examples. Nevertheless, what makes our approach different from the existing ones is the way we fuse the decisions of BNCs. In our approach, we train one BNC for each expert or for each data collection site. Then we train a fuser agent, using a reinforcement learning method, for attentive fusion of BNCs decisions. By *Attentive* we mean that the decision fuser learns to sequentially consult with a subset of BNCs for each sample.

2 Bayesian Network Classifier for Large Feature Spaces

Classification aims at accurately predicting the class label given attributes (features). In probabilistic framework, it is equivalent to estimating conditional likelihood $P(l_i|\mathcal{F}_i)$ where $l_i \in \{l_1, ..., l_w\}$ and $\mathcal{F}_i = \{f_i^1, ..., f_i^{n-1}\}$ are the class label and features representing sample $i(i = 1, ..., N)$ respectively. It is well-known that given a large dataset, it is possible to learn a BN exhibiting a close approximation of the underlying joint distribution. By having a joint distribution, it is possible to calculate any other conditional distribution such as conditional likelihood of a class label. Therefore, for classification, we first make the structure of a BN then train it for estimating the joint probability distribution by maximizing log-likelihood. Finally, the BN is used for calculating the conditional likelihood. However, in practice, this approach usually leads to inaccurate classifiers. As it is shown in [3], this is usually due to the fact that the likelihood term is mainly dominated by the joint distribution of features rather than the conditional likelihood. Technically, the log-likelihood function can be written as:

$$LL(B|TD) = \sum_{i=1}^{N} \log(P_B(l_i|f_i^1, ...f_i^{n-1})) + \sum_{i=1}^{N} \log(P_B(f_i^1, ..., f_i^{n-1})) \qquad (1)$$

where $P_B(.)$ is the probability calculated using the BN and TD is the training data.

As previously mentioned, to have better BNCs we should optimize conditional log-likelihood (CLL). Given the correct network structure, the parameters that maximize log likelihood (LL) also maximize CLL. Nevertheless, usually in practice, the optimal structure is unknown and maximum LL estimation will not optimize CLL. In addition, optimizing CLL does not have a closed-form solution. Moreover, unlike LL, CLL is not a decomposable score, which hinders the process of structure and parameter learning for CLL maximization.

It is also notable that as the number of features grows, LL will be dominated by the second term; because the possible instantiations of $\{f^1, ..., f^{n-1}\}$ increases. Consequently, usually $P_B(f_i^1, ..., f_i^{n-1})$ decreases. Therefore LL maximization pays less attention to maximization of $P_B(l_i|\mathcal{F}_i)$. As a result, we prefer to make multiple BNCs, each with a limited number of features, and then fusing their decisions.

3 Proposed Method

3.1 Using Reinforcement Learning for Attentive Fusion

Here, we introduce the framework, called *Local Bayesian Network Experts Fusion* (LoBNEF), by which the decisions of multiple BNCs are attentively fused. Instead of following the common practices in the fusion literature, in LoBNEF we map the problem of decision fusion to a *Markov Decision Process* (MDP). Then we incorporate a reinforcement learning method to solve the MDP. The fusion learner is called the learning agent.

The corresponding MDP is defined by 4-tuple $\{\mathcal{S},\mathcal{A},\mathcal{T},\mathcal{R}\}$ in which $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition probability function and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the reward function.

Let denote the finite set of features of the classification task by $\mathcal{F}$ and assume that we have m BNCs; see Fig 1. Also let denote the subset of features that the j^{th} classifier has access to by $\mathbb{F}^j \subseteq \mathcal{F}$ $(j = 1, ..., m)$. BNC's decision vector (output) for the i^{th} sample is defined as $Dec_i^j = [P(L = l_1)|\mathbb{F}_i^j, ..., P(L = l_w)|\mathbb{F}_i^j]$ where $P(L = l_k|\mathbb{F}_i^j)$ is the probability of belonging sample i to class l_k in the view of j^{th} BNC. In our setting, as in [8], the learning agents state is the augmentation of decision vectors of those BNCs which the agent has consulted with so far. Those entries corresponding to BNCs which the agent has not consulted with are filled with null. Therefore, the agent's state S is $S = [s_1 s_2 ... s_m]$ where:

$$s_j = \begin{cases} Dec^j & \text{if } j \in \text{selected networks so for} j = 1, ..., m \\ NULL_{1\times w} & \text{Otherwise} \end{cases}$$

The initial state of the agent is $S_0 = NULL_{1\times lw}$.

Action set $\mathcal{A}$ is the superset of two sets of actions: *Consulting Actions* (C) and *Declaration Actions* (D); i.e. $A = C \cup D$. A consulting action for sample i is asking a BNC its Dec_i^j while a declaration action means assigning a class label to the corresponding sample.

Finally, the reward function is designed to guide the learning agent to assign correct classes with minimum number of consultations (i.e. using minimum number of BNCs). Less consultation is of interest because each of them requires running an inference algorithm over the corresponding BNC which can be computationally expensive especially when the number of features grows. The reward function is:

$$R = \begin{cases} \text{High reward} & \text{correct class is assigned} \\ \text{High punishment} & \text{wrong class is assigned} \\ \text{Low punishment} & \text{consulted with a BNC} \end{cases}$$

So far, the problem of fusion of different BNCs is transformed to an MDP. Here, we are facing with an MDP with continuous state space and discrete actions. There exists a set of reinforcement learning methods to solve this MDP. We

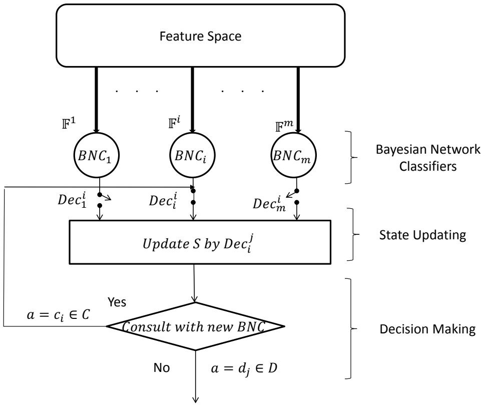


Fig. 1. A cycle of attentive fusion for sample i

use Bayesian Q-learning approach [9] because of its uncertainty handling and flexibility in generating prototypes.

Each episode starts by a query and ends by announcing the decided class label of query. During each episode of learning the learner starts from initial state S_0. Then, at each step it takes a consulting action and asks BNC its decision vector about the current sample. Finally, at the end of each episode, the learner takes a declaration action and assigns a label to the corresponding sample. Fig. 1 illustrates a schematic diagram of decision making in each episode. After each decision, either consultation or declaration, the agent receives the corresponding reward and updates its value function using Bayesian RL [9]. This process is iterated over samples several epochs until the process converges. To avoid overfitting, during each epoch the parameters of learner are saved. Then, the best set of parameters is chosen based on the CCR on the evaluation data.

3.2 Local Bayesian Network Learning

As shown in Fig. 1, we need a set of BNCs to attentively fuse their decisions. As we do not have a set of experts to build a BNC for each, we have two options; making BNCs for subsets of training data, constructing BNCs for subsets of features, or a combination of these. The second approach is closer to the real cases. In addition, it results in constructing fitter BNCs because of having a smaller number of features for each; see Section 2. We use the method proposed by Mirian et al. [8] to form the feature subsets.

In order to find the structure of Bayesian networks we use TAN algorithm [3]. TAN augments a naive Bayes structure by adding an additional parent for each node beside the class node. It is proven that this method constructs the optimal tree-augmented network that maximizes the likelihood function.

4 Experiments and Results

In this section we empirically evaluate the performance of LoBNEF both on hand-made and real datasets. In all experiments, 80% of data is used for training and evaluation; i.e. CV-5. The evaluation data is 10% of the training data, selected randomly. Also the number of epochs is 30 for all experiments.

4.1 Experiments on Simulated Data

The goal of this simulation is to show that our method can compete with the optimal-structure classifier and beat its competitors; namely TAN and naive Bayes. We first make a BN, see Fig 2.a, and generate 500 samples using it. We use the structure of this BN for the optimal-structure classifier as well. Then we train three BNCs each with five randomly selected features -out of the existing eight ones- and our method learns to attentively fuse their decision vectors. We repeat this process ten times by randomly changing the conditional probability table of the BN. For each set of data we train the competitors (TAN

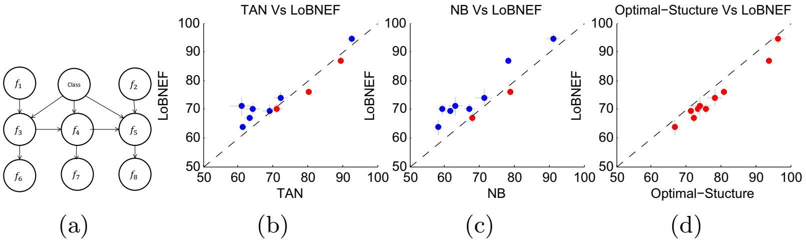


Fig. 2. Comparison of LoBNEF with TAN, Naive Bayes and optimal structure BNC. We varied the CPTs 10 times and used CV-5. Scatter plots show each classifier's CCR with one standard deviation. LoBNEF outperforms TAN (Naive Bayes) 7(8) times and is very close to the optimal-structure BNC.

and naive Bayes) and the optimal-structure classifier as well. Fig. 2.c(b) shows that our method beats TAN (naive Bayes) 7(8) times and is very close to the optimal-structure BNC. As discussed in Section 2, it is very probable that the structure learning methods fail to find the true structure of the BNC; especially in high dimensional cases. To simulate this case, we randomly removed from or added edges to the optimal structure and then trained the networks. We call these networks suboptimal-structure networks. The results show that drop in the average CCR of suboptimal-structure networks, in comparison to the optimal-structure one, is proportional to the probability of having extra or removed edges. For example, the average CCR falls below 68% when this probability is 10% - i.e. one edge is removed or is added in our simulation. It means, LoBNEF with CCR = 68.32% outperforms all of the suboptimal-structure networks. Note that in LoBNEF, each BNC has fewer features in comparison to the original BNC and making accurate BNCs for smaller number of features is more probable. In addition, attentive fusion of BNCs compensates lack of some features in each BNC. These two properties together, result in better performance for LoBNEF, in comparison to the suboptimal-structure networks.

4.2 Experiments on Real Datasets

We evaluated the performance of LoBNEF on 11 datasets from UCI repository [10]. Also, we used Mirian et al. [8] feature subset selection method to generate multiple feature subsets and the corresponding BNCs. For our experiments, we implemented LoBNEF, TAN and naive Bayes (NB) and compared the results with those of some benchmarking and state of the art methods from [7]; namely boosted augmented NB (BAN) [11], boosted Naive Bayes (BNB) [12], TAN with parameters optimized for conditional log likelihood (TAN-ELR) [1] and discriminative structure selection via CMDL score (BNC-2P) [4]. Continuous features were discretized using the supervised method proposed by Fayyad and Irani [13]. The average CCRs along with the standard deviations are shown in Table 1. Entries which are filled by N/A were not available for the corresponding datasets.

Table 1. Comparison between accuracy of TAN, NB, BAN, BNB, TAN-ELR, BNC-2P, LoBNEF. CV-5 used for all of data sets except Satimage that has pre-defined test and train sets. The fractions in parenthesise show the average number of consulting actions over the total number of BNCs. The best performances are in bold face.

Dataset	**TAN**	**NB**	**BAN**	**BNB**	**TAN-ELR**	**BNC-2P**	**LoBNEF**
Heart	86.27 ±0.23	82.59 ±0.91	84.44 ±0.78	84.07 ±0.41	81.53 ±0.98	83.33 ±1.10	**88.12(2.6/4)** **±0.64**
Hepatitis	82.50 ±1.85	85.00 ±1.43	**88.75** **±0.42**	87.50 ±0.85	86.98 ±0.85	87.50 ±2.12	84.06(3.3/5) ±1.31
Bupa	62.31 ±1.25	61.29 ±0.65	N/A	N/A	N/A	N/A	**71.61(2.4/5)** **±0.78**
Pima	70.84 ±0.27	72.18 ±0.87	75.73 ±0.61	76.06 ±0.86	76.16 ±0.32	73.94 ±0.84	**80.99(3/5)** **±0.33**
Ionosphere	80.21 ±1.54	**81.12** **±1.48**	N/A	N/A	N/A	N/A	79.99(2.4/4) ±1.74
Sonar	71.43 ±0.64	71.03 ±2.14	N/A	N/A	N/A	N/A	**84.23(2.3/4)** **± 1.21**
Glass	65.44 ±1.81	67.32 ±1.53	68.25 ±1.31	67.79 ±3.11	49.82 ±2.14	64.65 ±2.12	**69.91(1.9/4)** **±2.1**
Vehicle	68.46 ±1.67	68.54 ±1.54	67.24 ±2.1	67.54 ±1.78	**72.73** **±1.11**	65.48 ±1.83	67.01(2/4) ±2.11
Waveform	74.34 ±0.23	82.05 ±1.40	83.70 ±0.87	82.15 ±1.53	74.66 ±0.54	74.84 ±1.39	**85.12(3.1/5)** **±1.20**
Satimage	86.26 ±0.23	80.80 ±0.65	84.57 ±1.56	82.88 ±0.63	85.80 ±0.41	82.05 ±2.32	0.**87.01(3.7/5)** **±1.54**
Dermatology	84.13 ±1.45	85.12 ±1.32	N/A	N/A	N/A	N/A	**91.23(2.9/5)** **±2.1**

The ratios of average number of consultations over the constructed BNCs in LoBNEF are reported in the parenthesis. For learning the attentive fusion the reward function was 100 and -3000 for correct and wrong classification and the punishment was -5 for every consultation with BNCs. The table shows that LoBNEF significantly outperforms its competitors on 8 datasets. In addition, in contrast to BAN and BNB, LoBNEF does not use all BNCs for consultation and for every sample seeks advice from the most informative BNCs.

5 Conclusion and Future Works

We proposed a method, called LoBNEF, in that multiple BNCs are built to separately solve a classification task and then their outputs are attentively fused. The fusion is learned using a Bayesian reinforcement learning method. We demonstrated that our approach outperforms its competitors over some benchmarking datasets in addition to acting close to BNC with the true underlying structure in the simulations. Moreover, LoBNEF is more robust against error in constructing the network structure; compared to the case where a single network is used. In addition, while different domain experts may suggest different and even contradictory BNC structures for a single task, LoBNEF can make attentive use of all of them whenever they can improve the classification task; even in a portion of the problem domain. This attentive fusion is very efficient in terms of inference since it does not need to know every Bayesian networks decisions in order to assign a class to a sample. This is also highly desirable when the number of features increases and inference over Bayesian networks becomes a bottleneck.

This paper mainly studies the empirical side of LoBNEF and studying its theoretical properties is in our future research list. Extending our approach to other applications of BNs is among our current researches.

References

1. Greiner, R., Su, X., Shen, B., Zhou, W.: Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. Machine Learning 322, 297–322 (2005)
2. Su, J., Zhang, H., Ling, C., Matwin, S.: Discriminative parameter learning for bayesian networks. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1016–1023 (2008)
3. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning 29(2), 131–163 (1997)
4. Grossman, D., Domingos, P.: Learning bayesian network classifiers by maximizing conditional likelihood. In: Proceedings of the Twenty-First International Conference on Machine Learning (2004)
5. Carvalho, A., Roos, T., Oliveira, A., Myllymäki, P., et al.: Discriminative learning of bayesian networks via factorized conditional log-likelihood. Journal of Machine Learning Research (2011)
6. Geiger, D., Heckerman, D.: Knowledge representation and inference in similarity networks and bayesian multinets. Artificial Intelligence 82(1-2), 45–74 (1996)
7. Jing, Y., Pavlovi, V., Rehg, J.: Boosted bayesian network classifiers. Machine Learning 73(2), 155–184 (2008)
8. Mirian, M.S., Ahmadabadi, M.N., Araabi, B.N., Siegwart, R.R.: Learning active fusion of multiple experts' decisions: An attention-based approach. Neural Computation 23(2), 558–591 (2011)
9. Firouzi, H., Ahmadabadi, M., Araabi, B., Amizadeh, S., Mirian, M., Siegwart, R.: Interactive learning in continuous multimodal space: A bayesian approach to action based soft partitioning and learning. IEEE Transactions on Autonomous Mental Development (99) (2011)
10. Newman, D., Hettich, S., Blake, C., Merz, C.: Uci repository of machine learning databases. Department of Information and Computer Science. University of California, Irvine (1998)
11. Jing, Y., Pavlovi, V., Rehg, J.: Efficient discriminative learning of bayesian network classifier via boosted augmented naive bayes. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 369–376 (2005)
12. Ridgeway, G., Madigan, D., Richardson, T., Okane, J.: Interpretable boosted naive bayes classification. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 101–104. AAAI Press (1998)
13. Fayyad, U., Irani, K.: Multi-interval discretization of continuous-valued attributes for classification learning (1993)

Multiclass Penalized Likelihood Pattern Classification Algorithm

Amira Samy Talaat[1], Amir F. Atiya[2,*], Sahar A. Mokhtar[1], Ahmed Al-Ani[3], and Magda Fayek[2]

[1] Computers and Systems Department, Electronic Research Institute
{am_samy,sahar}@eri.sci.eg
[2] Department of Computer Engineering, Cairo University, Giza, Egypt
amir@alumni.caltech.edu, magdafayek@gmail.com
[3] Faculty of Engineering and Information Technology, Univ. of Technology, Sydney, Australia
ahmed@eng.uts.edu.au

Abstract. Penalized likelihood is a general approach whereby an objective function is defined, consisting of the log likelihood of the data minus some term penalizing non-smooth solutions. Subsequently, this objective function is maximized, yielding a solution that achieves some sort of trade-off between the faithfulness and the smoothness of the fit.

In this paper we extend the penalized likelihood classification that we proposed in earlier work to the multi class case. The algorithms are based on using a penalty term based on the K-nearest neighbors and the likelihood of the training patterns' classifications. The algorithms are simple to implement, and result in a performance competitive with leading classifiers.

Keywords: Multiclass, K-nearest neighbor, Penalized likelihood, Pattern classification, Posterior probability.

1 Introduction

The basic concept behind penalized likelihood is that a good model should possesses two essential properties: the goodness of fit and the smoothness of the fit [1], [2]. However, these two are primarily conflicting goals, and usually a trade-off that suits the given application is pursued. The penalized likelihood approach seeks to achieve that trade-off by defining an overall objective function consisting of the log-likelihood of the data minus a roughness measure, and subsequently maximizing this objective function. The likelihood function is a measure of the faithfulness of the fit, while the roughness function is a penalty term that penalizes non-smooth solutions. An example of the roughness function is the integral of the square of the second derivative of the function, leading to the following objective function (see [3]):

$$T = \log likelihood - \lambda \int f''^2(x)dx \tag{1}$$

Most of the penalized regression work focused on finding a complete functional formulation and the optimization is performed mostly in the Hilbert space (see [4]).

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 141–148, 2012.

In contrast to the regression framework, there is little work on extending it to the classification domain. For the classification problem the underlying function would then be the class posterior probabilities. These are the functions which we attempt to estimate and for which we impose smoothness. Among the works considering penalized likelihood classification is the work of O'Sullivan et al [5], which was subsequently analyzed and extended in many other studies (see [6-9]). A related approach is to consider the multinomial logistic regression case in Cawley et al [10]).

Atiya and Al-Ani [11] proposed a new model for penalized likelihood classification. The idea of their approach is to evaluate the posterior probabilities for the training and the testing points. They use as a measure of roughness the sum of square difference between the posterior of a point and that of its K nearest neighbors. However, their model applied only to two-class classification, and it was not clear how to extend that to the multi-class case, as a new derivation is needed. In this work we extend their work to the multi-class case. We derive two new algorithms for obtaining the estimates of the posterior probabilities. One of them is based on the gradient ascent algorithm, and the other is based on component-wise optimization.

2 The Proposed Method

Let $x_m \in R^L$ denote the feature vectors for pattern m, with $x_1, \ldots, x_M$ denoting the training patterns, and $x_{M+1}, \ldots, x_{M+N}$ denoting the test patterns. Let g be an index of the class number with G being the number of classes: $g = 1,2, \ldots G$ and let $P_{m1}, P_{m2,\ldots\ldots,}P_{mG}$ be posterior probabilities of classes $1,2, \ldots G$ for pattern m ($P_{mg} \equiv P(g|x_m)$). Let $\mathrm{y}_{m\mathrm{g}}$ be the class membership for training pattern x_m which is defined as $\mathrm{y}_{mg} = 1$ if it belongs to class g and equals zero otherwise.

The purpose of the proposed method is to estimate the posterior probabilities P_{mg}, both for the training set and the test set. Knowing the posterior probabilities will automatically determine the classification of the patterns. The posterior probabilities are obtained by defining the penalized likelihood function and subsequently maximizing it, leading to the proposed iterative algorithms.

Knowing that

$$\sum_{g=1}^{G} P_{mg} = 1 \quad \text{then} \quad P_{mG} = 1 - \sum_{g=1}^{G-1} P_{mg} \tag{2}$$

the likelihood of the data is given by

$$L = \prod_{m=1}^{M} \left(\prod_{g=1}^{G-1} P_{mg}^{y_{mg}}\right)\left(1 - \sum_{g=1}^{G-1} P_{mg}\right)^{y_{mG}} \tag{3}$$

Denote by $\mathcal{K}(x_m)$ as the set of K-nearest neighbors of point x_m (their indexes). We define a roughness function based on the square differences of the posteriors of neighboring data points. Specifically, Roughness measure is given by:

$$S = \frac{1}{\mathrm{K}} \sum_{m=1}^{M+N} \sum_{m' \in \mathcal{K}(x_m)} \sum_{g=1}^{G-1} (P_{mg} - P_{m'g})^2 \tag{4}$$

We define our overall objective function as a combination of (3),(4):

$$\mathrm{J} = \log(L) - \lambda S \tag{5}$$

$$J = \sum_{m=1}^{M}\left[\sum_{g=1}^{G-1} y_{mg} \log\left(P_{mg}\right) + y_{mG}\, log\left(1 - \sum_{g=1}^{G-1} P_{mg}\right)\right] - \frac{\lambda}{K}\sum_{m=1}^{M+N}\sum_{m' \in \mathcal{K}(x_m)}\sum_{g=1}^{G-1}\left(P_{mg} - P_{m'g}\right)^2 \tag{6}$$

The first term in the penalized log-likelihood J focuses on the goodness of fit aspect. It measures how well the considered $P_{mg}{}'s$ fit the observed data (i.e. the given class memberships). The second term's purpose is to penalize the roughness of the underlying posterior function. A posterior surface where its values for neighboring points are close (i.e. having low S) will generally be smooth, and conversely a high S is indicative of a rough surface. The goal is to find the posterior probabilities that maximize the penalized log-likelihood J. We will therefore achieve a compromise between faithfully respecting the class memberships of the training data and the smoothness property of the posterior surface, with λ being the parameter that controls the degree of smoothness. Note that the testing patterns are also used in the expression for the smoothness function (the summation in S is over the entire data set). Even though they do not carry classification labels, they could be helpful in bridging the gaps between the training patterns to achieve a smoother fit. On the other hand, the summation for the log-likelihood function is over only the training set. The reason is that class labels are known only for the training set, but not for the test set. We emphasize that the labels of the test data were and should never be used in the classifier design, in order to guarantee fair testing.

3 The Proposed Algorithm

3.1 Initial $\mathbf{P_{mg}}$ Choice:

The initial choice of P_{mg} is selected as $m = 1, \dots, M, P_{mg} = y_{mg}$, and for $m = M + 1, \dots, M + N, P_{mg} = \frac{1}{G}$. The goal is to solve the following maximization problem:

Maximize J (given by (6)) w.r.t. the variables: P_{mg}, s.t. $0 \le P_{mg} \le 1$, $m = 1, \dots, M + N$. Thus, we are dealing with a constrained maximization optimization problem, where the constraints are just bounds for the variables P_{mg}. We propose two algorithms for solving the above mentioned optimization problem. The first algorithm is based on gradient ascent (Method 1), and the second algorithm is based on cycling through all variables, each time optimizing w.r.t. only one of the variables through a line search (Method 2). The algorithms are described as follows:

3.2 Method 1

By using Gradient Ascent method, we start with initial choice for $P_{mg}{}'s$ as described in section 3.1, then update $P_{mg}{}'s$ in the direction of the gradient and repeat until convergence to a good solution.

$$P_{mg}(New) = P_{mg}(Old) + \eta \frac{\delta J}{\delta P_{mg}} \tag{7}$$

To get $\frac{\delta J}{\delta P_{mg}}$ equation (6) is partitioned to two terms $Term1$ (the likelihood part) and $Term2$ (the roughness term)

$$\frac{\delta Term1}{P_{nj}} = \left(\frac{y_{nj}}{P_{nj}} + \frac{y_{nG}}{\sum_{g'=1}^{G-1} P_{ng'} - 1}\right) \tag{8}$$

$$\frac{\delta Term2}{P_{nj}} = 2\frac{\lambda}{K}\left[(K + K_s)P_{nj} - \sum_{n' \epsilon \mathcal{K}(x_n)} P_{n'j} - \sum_{n' \epsilon \mathcal{S}(x_n)} P_{n'j}\right] \tag{9}$$

where K ≡ number of neighbors used in KNN, K_s ≡ size of $\mathcal{S}(x_n)$, and $\mathcal{S}(x_n)$ is the set of points for which x_n is a K-nearest neighbor. For any $1 \leq n \leq M + N$:

$$\frac{\delta J}{\delta P_{nj}} = \frac{\delta Term1}{P_{nj}} - \frac{\delta Term2}{P_{nj}} \tag{10}$$

Get P_{mg} for $g = 1\ to\ G - 1$, apply equation (2) to get the dependent class G probabilities P_{mG}, apply gradient (steepest) ascent from equation (7), and Truncate if P_{mg} goes out of the constraint box:

$$\text{Set } P_{mg} = 1 \text{ if } P_{mg} > 1 \text{ and set } P_{mg} = 0 \text{ if } P_{mg} < 0 \tag{11}$$

3.3 Method 2

This is the generalization of the method proposed in [11]. We start with an initial choice for $P_{mg}'s$ like described in section 3.1. While the change in the posteriors (the $P_{mg}'s$) between the current and previous iteration is greater than a certain threshold, for all patterns *m* do the following:

(a) If $y_{m\mathrm{g}} = 0$ and $y_{m\mathrm{G}} \neq 1$ which means that the training pattern belongs to a different class than current considered class *g* and the dependent last class *G*, then set:

$$P_{mg} = \bar{P}_{mg} = \frac{1}{K+K_s}\left[\sum_{m' \in \mathcal{K}(x_m)} P_{m'g} + \sum_{m' \in \mathcal{y}(x_m)} P_{m'g}\right] \tag{12}$$

where K the number of nearest neighbors is, $\mathcal{y}(x_m)$ is the set of data points for which x_m is one of the K-nearest neighbors, and K_s is the size of set $\mathcal{y}(x_m)$. Thus $\bar{P}_{mg}$is the mean of the values of $P_{m'g}$ for some sort of neighborhood of points around x_m.

(b) If $y_{mg} = 1$, which means that the training pattern belongs to the same current class *g* then set:

$$P_{mg} = \frac{\bar{P}_{mg}}{2} + \sqrt{\left(\frac{\bar{P}_{mg}}{2}\right)^2 + \beta} \tag{13}$$

where $= \frac{1}{2(K+K_s)}$, and $\bar{P}_{mg}$ is given as in (12).

(c) If $y_{mG} = 1$, which means that the training pattern belongs to the dependent final class G, then set:

$$P_{mg} = \bar{P}_{mg} + \left[\frac{1-\sum_{g=1}^{G-1} \bar{P}_{mg}}{2(G-1)}\right] - \sqrt{\left(\frac{1-\sum_{g=1}^{G-1} \bar{P}_{mg}}{2(G-1)}\right)^2 + \frac{\beta}{(G-1)}} \tag{14}$$

Note that $1-\sum_{g=1}^{G-1}\bar{P}_{mg}$ here represents the neighborhood average of the posteriors of the dependent class G.

(d) If x_m is a test pattern then set $P_{mg}=\bar{P}_{mg}$. Finally apply the constraints on the probabilities (2), (11).

Essentially, what this algorithm performs is iterated local averaging of the posteriors (to obtain $\bar{P}_{mg}$), and combining the resulting average in some way with the class membership (i.e. y_{mg}) of the considered pattern (if known), through (12), (13),(14).

The iteration cycles change the posteriors. Once the algorithm converges, we use the obtained final values of the $P_{mg}'s$ as the estimated posteriors of data points (whether training data or testing data). Recalling that $P_{mg}\equiv P(g|x_m)$ denotes the posterior probability for class g, then the final classification of a data point is estimated as class g if P_{mg} is the highest.

The new proposed methods Method1 and Method2 are updated pattern by pattern, which means that inside each cycle the P_{mg}'s of only one pattern are updated at a time. Since the dependent class G is dealt with in a special way in the above algorithms, we rotate this selected special class over all problem classes from $g=1\ to\ G$, and apply the algorithm again. We end up with G solutions P_{mg}. To select which solution set is the best, we select the one that gives best accuracy over the training set. It is this one that will be selected, and applied to the test set.

4 Simulation Results

We have compared the performance of the proposed method to that of the following well-known classifiers in table 1, together with their abbreviations. We tested methods on real-world pattern classification problems, from the UCI repository [12].

Table 1. The classification models used in the comparison, and their abbreviations

Classifier	Abbreviation
Mainfold Parzen Window Bayes classifier	Parzen
Support vector machines (linear)	SVML
Support vector machines (RBF kernel)	SVMR
K-nearest neighbor	KNN
Neighborhood components analysis	NCA
NaiveBayes kernel Classifier	NBk
Decision Tree Classifier	DT

- SVM: We tested linear SVM (SVML) and SVM RBF (SVMR). The two methods implemented using LIBSVM software [13] using the default values for the parameters.
- Parzen [14]: We use manifold Parzen software [15], the value of sigma, the standard deviation, of the Parzen probability density function is selected as 1.
- K-nearest neighbor classifier: We used the value K = 9.

- NCA [16] by NCA software [17], we selected the value of K=9.
- NBk: Create NB object by fitting training data kernel smoothing density estimate.
- DT: decision tree with binary classification splits; the model of trained tree used to predict classes.

Table 2. The datasets used to evaluate the performance of the classifiers

Database Name	Patterns	Attributes	Classes
Teaching Assistant Evaluation (tae)	151	6	3
Statlog (Heart)	270	14	3
Breast (Tissue)	106	11	6
Contraceptive Method Choice (cmc)	1473	10	3
Dermatology (derm)	358	35	6
Iris	150	5	3
Ecoli	336	8	8
Post-Operative Patient (Patient)	87	9	3
Vertebral Column (verteb)	310	7	3

Table 2 summarizes the characteristics of the datasets used in this paper. Patterns or attributes that consist of missing values were removed from the datasets. We used 10 fold cross validation for training and testing. This is because some of the data sets are small, so a hold-out test would yield a smaller size test set, so we opted for K-fold method of rotating the test set. We performed 10 runs for each method. Then we average the classification accuracies on the test sets of the 10 runs.

Using experiment on other data we found that for the proposed method the best K value equals 9, and the best λ equals 1. The algorithm for Method 2 described in Section 3.3 needed only 1 or 2 iterations to converge for all datasets, while the algorithm Method 1 described in Section 3.2 we need maximum 9 iterations to converge for all datasets.

Table 3. Average classification accuracy of the competing methods

	M1	**M2**	**KNN**	**NCA**	**SVML**	**SVMR**	**Parzen**	**NBk**	**DT**
tae	**68.88**	43.13	39.13	45.08	50.25	52.29	54.92	51.71	57.00
tissue	82.09	63.91	49.09	52.45	92.27	14.82	13.91	86.55	**96.27**
heart	**72.59**	68.89	62.59	62.22	69.63	46.67	45.93	67.41	61.48
cmc	**56.76**	55.61	53.30	46.64	50.59	56.35	53.44	52.68	50.71
derm	87.98	89.92	83.82	78.79	**95.82**	92.18	91.33	90.77	94.42
verteb	**90.00**	87.74	81.94	81.29	85.16	48.39	79.35	80.32	79.03
patient	**87.22**	70.83	66.39	66.25	68.47	70.83	69.72	70.83	65.42
ecoli	**89.92**	89.92	86.93	75.04	81.27	75.61	42.59	83.69	81.01
iris	**99.33**	97.33	96.67	96.00	97.33	97.33	92.00	96.00	94.67
Av.	**81.64**	74.14	68.87	67.09	76.76	61.61	60.35	75.55	75.56

Table 3 presents the average classification accuracy of the competing methods with average of each method over all datasets used. For the given nine datasets the best classifier is marked bold, One can observe that Method1 gives the best accuracy for seven datasets , while DT, SVML gives the best accuracy for tissue, derm datasets respectively. Method2 has average accuracy better than KNN, NCA, SVMRBF, Parzen but worse than SVML, NBK, DT.

Overall, the average accuracy of Method1 is the best among all competing methods. It is therefore considered a competitive classification method and should be tested along with other leading classifiers for any classification task.

5 Conclusion

In this paper we have developed two new classification methods based on the penalized likelihood concept. The proposed method was compared with several existing classification methods. It gave a best performance over all models. We therefore believe that the proposed approach offers superior performance, and as such it should be one of the major contenders to be tested or used in multi-class classification task.

References

1. Green, P.: Penalized Likelihood. In: Encyclopedia of Statistical Sciences Update, vol. 3. John Wiley Publishing, New Jersey (1999)
2. Gu, C., Kim, Y.J.: Penalized Likelihood Regression: General Formulation and Efficient Approximation. Can. J. Stat. 30, 619–628 (2002)
3. Green, P.J., Silverman, B.W.: Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach. Chapman and Hall, London (1994)
4. Wahba, G.: Spline Models for Observational Data. SIAM, Philadelphia (1990)
5. O'Sullivan, F., Yandell, B., Raynor, W.: Automatic Smoothing of Regression Functions in Generalized Linear Models. J. Am. Stat. Assoc. 81, 96–103 (1986)
6. Gu, C.: Cross-validating Non-Gaussian Data. J. Comput. Graph. Stat. 1, 169–179 (1992)
7. Lu, F., Hill, G.C., Wahba, G., Desiati, P.: Signal Probability Estimation with Penalized Likelihood Method on Weighted Data. Department of Statistics, University of Wisconsin, Technical Report No. 1106 (2005)
8. Wahba, G.: Soft and Hard Classification by Reproducing Kernel Hilbert Space Methods. Proceedings of the National Academy of Sciences 99, 16524–16530 (2002)
9. Wahba, G., Gu, C., Wang, Y., Chappell, R.: Soft Classification, a.k.a. Risk Estimation, via Penalized Log Likelihood and Smoothing Spline Analysis of Variance. Department of Statistics, University of Wisconsin, Technical Report No. 899 (1993)
10. Cawley, G., Talbot, N.L., Girolami, M.: Sparse Multinomial Logistic Regression via Bayesian L1 Regularisation. Adv. Neural Inf. Process. Syst. 19, 209–216 (2007)
11. Atiya, A.F., Al-Ani, A.: A Penalized Likelihood Based Pattern Classification Algorithm. Pattern Recogn. 42, 2684–2694 (2009)
12. UCI Machine Learning Repository (2012), `http://archive.ics.uci.edu/ml/`
13. Chang, C.C., Lin, C.J.: LIBSVM toolbox (2012), `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

14. Vincent, P., Bengio, Y.: Manifold Parzen Windows. Adv. Neural Inf. Process. Syst. 15, 825–832 (2003)
15. Paris, S.: Parzen Windows Estimator Classifier (2008), `http://www.mathworks.com/matlabcentral/fileexchange/17450-parzen-classifier`
16. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood Components Analysis. Adv. Neural Inf. Process. Syst. 17, 513–520 (2004)
17. Maaten, L.V.D.: NCA Toolbox for Dimensionality Reduction (2010), `http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html`

Self-Organising Maps for Classification with Metropolis-Hastings Algorithm for Supervision

Piotr Płoński and Krzysztof Zaremba

Institute of Radioelectronics, Warsaw University of Technology,
Nowowiejska 15/19,00-665 Warsaw, Poland
{pplonski,zaremba}@ire.pw.edu.pl

Abstract. Self-Organising Maps (SOM) provide a method of feature mapping from multi-dimensional space to a usually two-dimensional grid of neurons in an unsupervised way. This way of data analysis has been proved as an efficient tool in many applications. SOM presented by T.Kohonen originally were unsupervised learning algorithm, however it is often used in classification problems. This paper introduces novel method for supervised learning of the SOM. It is based on neuron's class membership and Metropolis-Hastings algorithm, which control network's learning process. This approach is illustrated by performing recognition tasks on nine real data sets, such as: faces, written digits or spoken letters. Experimental results show improvements over the state-of-art methods for using SOM as classifier.

Keywords: Self-Organising Maps, Classification, Supervised learning, Metropolis-Hastings algorithm.

1 Introduction

Self-Organising Map (SOM) is a neural network presented in 1982 by T.Kononen [11]. Due to human readability of the model, easy implementation and fast learning, SOM gained great popularity in many data analysis problems [13]. SOM was originally presented as an unsupervised algorithm, however there are extensions that enable to use SOM as a classifier. They can be generally divided into three groups.

The first group of methods is based on class membership of each neuron. In this approach, SOM is first learned in unsupervised manner. After training, class membership is found for each neuron, based on sample's class label. Class membership can be crisp or fuzzy [8], [9], [18], [20]. In the testing phase, the simplest approach predicts the class based on the class of winning neuron - so-called 'winner-takes-all method' (WTA). There are also more sophisticated methods based for example on k-Nearest Neighbour rule or interpretation of weights[21].

The second approach combines class vector in binary coded manner with attribute vector during learning process. In the testing phase, only the attribute

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 149–156, 2012.

vector is presented to the SOM. Sample's class is denoted, based on neurons weights corresponding to the class. There are several approaches for doing this [2], [9], [19], [16], [22]. However, the [16] algorithm seems to present the most generalized approach from those.

The third technique for using SOM as a classifier is to use as many SOM networks as number of classes. This approach is well known from Learning Vector Quantization (LVQ) algorithm [12]. In the simplest way, each network is trained on samples from corresponding class [4], [7]. In more complex approach [12], the network is trained on samples from both corresponding and other classes.

Method presented in this paper combines first and third approach.It uses Metropolis-Hastings (MH) algorithm [6], [15] and class membership of neurons to control neurons participation in the training process. The MH is well know from Simulated Annealing (SA) [10] algorithm. There were several attempts to use MH [17] or SA [3], [5] in SOM. However, they were focused on weights optimization rather than boosting SOM's classification performance.

2 Methods

Let's denote data set as $D = \{(\boldsymbol{x_i}, c_i)\}$, where $\boldsymbol{x_i}$ is an attribute vector, $\boldsymbol{x} \in \mathcal{R}^d$ and c_i is a discrete class number of i-th sample, $i = [1, 2, ..., N]$ and $c = [1, 2, ..., C]$. Sometimes the class number will be encoded as a binary vector and denoted as $\boldsymbol{y_i}$, where $\boldsymbol{y_{ij}} = 1$ for $j = c_i$ and $\boldsymbol{y_{ij}} = 0$ otherwise.

2.1 Unsupervised Learning SOM Algorithm

Herein, we used SOM as a two-dimensional grid of neurons. Each neuron is represented by a weight vector W_{pq}, where (p, q) are indexes of the neuron in the grid. In the learning phase all samples are shown to the network in one epoch. For each sample we search for a neuron which is closest to the i-th sample. The distance is computed by:

$$Dist_{train}(D_i, W_{pq}) = (\boldsymbol{x_i} - W_{pq})^T(\boldsymbol{x_i} - W_{pq}). \tag{1}$$

The neuron (p, q) with the smallest distance to i-th sample is called the Best Matching Unit (BMU), and we note its indexes as (r, v). Once the BMU is found, the weight update step is executed. The weights of each neuron are updated with the following formula:

$$W_{pq}(t+1) = W_{pq}(t) + \eta(W_{pq}(t) - \boldsymbol{x_i}), \tag{2}$$

where t is an iteration number and η is a learning coefficient. It can be written as $\eta = \mu\tau$, where μ is the size of the learning step and τ is the neighbourhood function. Learning step size is decreased between consecutive epochs, so that network's ability to remember patterns is improved. It is described by $\mu = \mu_0 exp(-e\lambda_\mu)$, where μ_0 is the initial step size, e is the current epoch number and λ_μ is responsible for regulating the speed of the decrease. Neighbourhood function controls changing of the weights with respect to the distance to the BMU. It is noted as $\tau(r,v,p,q) = exp(-\alpha((r-p)^2+(v-q)^2))$, where α describes the neighbourhood function width. This parameter is increasing during learning $\alpha = \alpha_0 exp(-(e_{stop} - e)\lambda_\alpha)$ - it assures that neighbourhood becomes narrower during training. Network is trained till chosen number of learning procedure epochs e_{stop} is exceeded.

2.2 SOM-WTA

From the first group of methods we will use SOM in WTA configuration (SOM-WTA). After unsupervised training process, where BMU for each sample is found, the class membership for each neuron is computed. BMU contains class number of the matching samples. After presentation of all the samples, each neuron's class membership is decided based on major class number. In the testing phase, the class of an input sample is assigned based on the class of the computed BMU. The main disadvantage of this method are so-called 'empty neurons', when neuron has never been selected as BMU during training but is selected in the testing[21].

2.3 SOM-LASSO

The second approach used in this paper is the so-called 'Learning Associations by Self-Organisation' (SOM-LASSO), first described in [16]. During the learning phase, additionally to attributes it takes into consideration the class vector $\boldsymbol{y_i}$. Each neuron contains part of weights corresponding to the attributes W^x_{pq} and a class vector W^y_{pq}, so $W_{pq} = [W^x_{pq}; W^y_{pq}]$. The measure of the distance used during training is computed by:

$$Dist_{train}(D_i, W_{pq}) = (\boldsymbol{x_i} - W^x_{pq})^T(\boldsymbol{x_i} - W^x_{pq}) + (\boldsymbol{y_i} - W^y_{pq})^T(\boldsymbol{y_i} - W^y_{pq}). \quad (3)$$

The rest of the training process is the same as in original SOM. In testing, the exploitation phase is performed, where only the part with attributes is presented to the network. The BMU is found by computing a distance between an attribute input vector and an attribute part of the weights, using the following formula:

$$Dist_{test}(D_i, W_{pq}) = (\boldsymbol{x_i} - W^x_{pq})^T(\boldsymbol{x_i} - W^x_{pq}). \quad (4)$$

For the tested sample, the designated class corresponds to position of maximum value in the part which codes class information W^y_{pq} in BMU weights.

2.4 SOM-SNEC

The third method uses separate network for each class, we called it SOM-SNEC. Each network is learned only with samples from the corresponding class in an unsupervised manner. In the testing process, the BMU is computed in each network. Sample's class is designated from the network with the closest BMU. The main disadvantage of this method is that it losts possibility to visualize all samples on a single map.

2.5 Proposed Method (SOM-MH)

In this method, neuron's class membership is described by probability. We note $P_{pq}(h)$ as probability of neuron's membership in class number h, where (p, q) are neuron's indexes. For each training iteration[1] only selected group of neurons will take part in the training. Selection is described by a matrix T, where $T^i_{pq} = 1$ means that neuron (p, q) will participate in learning using i-th sample, $T^i_{pq} = 0$ otherwise. Neurons are selected in two steps. First choose neurons having maximum probability for the class matching the class c_i of the input sample:

$$T^{i(1)}_{pq} = \begin{cases} 1 & \text{if } \arg\max_h(P_{pq}(h)) = c_i; \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

In the second step, remaining neurons are considered, with $T^{i(1)}_{pq} = 0$. The decision on joining the training with i-th sample is taken upon MH algorithm. The probability of joining is computed using following equation:

$$J^i_{pq} = 1 - exp(-\rho P_{pq}(c_i) e_{stop}/e), \tag{6}$$

where ρ is the parameter that controls the number of neurons selected additionally to learning in the MH step, $\rho \in [0, 1]$. The fact that number of epochs e is presented in eq.(6) ensures that neurons added during MH step will be selected less frequently at the end of learning process than at its beginning. This can be interpreted as a hesitation of the neuron, which decreases during the training. Whether the MH decision will be positive, we draw random number a from an uniform distribution, $a \in [0, 1]$. The neuron will be added to the training group if a is smaller than J^i_{pq}:

$$T^{i(2)}_{pq} = \begin{cases} 1 & \text{if } a < J^i_{pq}; \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

This procedure is repeated for each sample. The final decision on neuron selection is a logical 'or' of the decisions $T^i_{pq} = T^{i(1)}_{pq} \vee T^{i(2)}_{pq}$.

After each epoch new probabilities are updated. During training for each i-th sample the neighbourhood value τ_i is added to the neuron's probability of membership in a given class:

$$P'_{pq}(h) = \sum_i^N T^i_{pq} \tau_i, \text{ for } h = c_i. \tag{8}$$

The neighbourhood value τ_i represents the belonging of the neuron to the input sample's class. After all iterations in a given epoch, the probability are normalized and updated with formula:

$$P_{pq}(h) = \frac{P'_{pq}(h)}{\sum_{j=1}^C P'_{pq}(j)}. \tag{9}$$

[1] One iteration is a showing to the network one sample. One epoch is a showing to the network all samples.

3 Results

At the beginning we will present properties of proposed SOM-MH method, then we will compare it to the SOM-WTA, SOM-LASSO, SOM-SNEC and LVQ methods. The comparison is made on 9 real data sets. We used data sets 'Wine', 'Ionosphere', 'Iris', 'Isolet', 'Digits', 'Sonar', 'Spam', 'Pima' from the 'UCI Machine Learing Repository' [2] [1], and set 'Faces' are from the 'The ORL Database of Faces'[3]. In all experiments we used following parameters values: $e_{stop} = 200$, $\mu_0 = 0.1$, $\lambda = 0.0345$, $\alpha_0 = 0.1$, $\lambda_\alpha = 0.008$. All variants of SOM algorithms were implemented by authors in Matlab. The LVQ algorithm was used from Matlab Neural Networks Toolbox with default learning parameters and number of epochs e_{stop}.

Table 1. Description of data sets used to test performance and parameters of networks. Single net size was used for methods SOM-WTA, SOM-LASSO and SOM-MH, multiple nets size is for SOM-SNEC and LVQ. (*) In 'Isolet' and 'Faces' data sets, the number of attributes was reduced with PCA.

	Train examples	Test examples	Attributes	Classes	Single net size	Multiple nets size	MH ρ
Faces	320	80	50*	40	15x16	2x3	0.005
Ionosphere	280	71	34	2	6x8	4x6	0.005
Iris	120	30	4	3	6x6	3x4	0.25
Isolet	6237	1560	100*	26	12x13	2x3	0.75
Digits	4496	1124	64	10	15x16	4x6	0.5
Wine	142	36	13	3	6x6	3x4	0.2
Pima	614	154	8	2	12x12	8x9	0.25
Sonar	166	42	60	2	8x9	6x6	0.1
Spam	3680	921	57	2	12x12	8x9	0.75

To show SOM-MH algorithm properties, we learned 7x7 network with 'Iris' data set. Fig.1a presents network with neurons assigned to one of the three classes. Fig.1b presents cumulative number of positive MH decisions taken for each neuron during the whole training. We can observe that neurons which lay on the border between the different classes have higher number of positive MH decisions than neurons which have neighbour neuron from the same class. The highest number of positive MH decisions are for neuron which lay in the border of the three classes. Fig.1c presents number of positive MH decisions for network in each epoch for MH parameter $\rho = 0.5$. It can be observed that number of positive MH decisions are decreasing during learning, which can be interpreted as making the network more confident.

Network sizes used for each data sets are presented in Table 1. For each method, the total number of used neurons are the same. For SOM-WTA, SOM-LASSO,

[2] http://archive.ics.uci.edu/ml/

[3] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

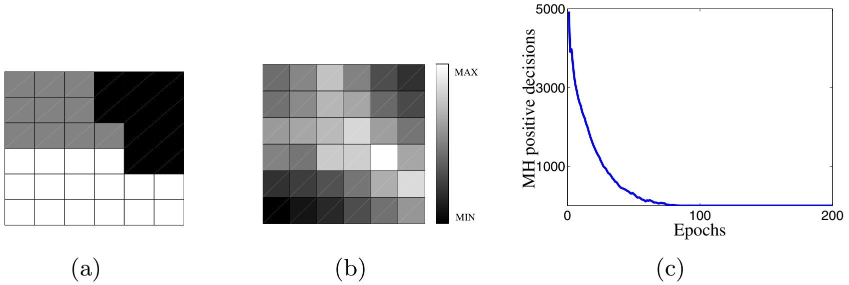


Fig. 1. Properties of SOM-MH network tested on 'Iris' data set, (a) network with neuron color presenting class membership, (b) network with neuron color presenting number of positive MH decisions taken during training, (c) number of positive MH decisions of all network taken in each training epoch.

SOM-MH for all the classes used a single network. For SOM-SNEC and LVQ, there are multiple networks, one for each class (and hence different column in Table 1). For SOM-MH the parameter ρ must be tuned. We checked several values of ρ, $\rho = \{1, 0.75, 0.5, 0.25, 0.2, 0.15, 0.1, 0.05, 0.005\}$ and for each data set an optimal value was selected by cross-validation. Selected ρ values are presented in Table 1. For each data sets we made 10 repetitions to avoid effect of local minima. As an accuracy measure, we take the percentage of incorrect classifications. The mean results for all the methods for training subsets are presented in Table 2.

The poorest accuracy on almost all data sets was obtained by SOM-WTA method. This was expected, as this method does not use the information about sample's class during the tuning of the weights. However, this method was better than SOM-LASSO on the 'Faces' training set. Poor accuracy of SOM-LASSO on this set can be explained by comparable lengths of attribute and class vectors. The best accuracy on this set was obtained by SOM-SNEC method, which was also the best method on 'Isolet' data sets. On 'Pima' data set the LVQ method

Table 2. Percent of incorrect classification on testing subsets for the SOM-WTA, SOM-LASSO, SOM-SNEC, SOM-MH and LVQ methods. Results are mean and σ over 10 runs.

	SOM-WTA	SOM-LASSO	SOM-SNEC	SOM-MH	LVQ
Faces	28.88±4.35	35.75±5.11	**3.75±2.04**	5.25±2.99	6.5±3.05
Ionosphere	14.23±4.06	13.66±4.2	10.85±3.45	**10.42±3.65**	13.1±3.93
Iris	6.67±4.97	6±3.06	3.33±2.22	**2±1.72**	6±4.66
Isolet	21.5±1.74	8.36±0.61	**5.96±0.45**	6.83±0.8	7.6±0.48
Digits	6.27±0.8	6.29±0.64	3.16±0.44	**3.02±0.44**	17.94±2.44
Wine	6.94±4.77	4.17±4.39	3.33±2.55	**2.74±2.27**	4.17±2.36
Pima	28.05±4.59	24.55±2.41	26.69±3.39	22.4±3.36	**21.56±3.77**
Sonar	36.19±6.99	24.76±5.96	24.05±4.69	**23.81±6.04**	26.67±6.53
Spam	16.35±1.02	12.74±1.25	12.42±1.17	**11.77±1.3**	37.74±1.34

gives the best performance. On all other sets, the SOM-MH method gives the lowest incorrect classifications. If the comparison is made only for methods that use single SOM network, SOM-MH is significantly better than SOM-WTA and SOM-LASSO on all data sets. SOM-SNEC has similar results to SOM-MH. However, by using SOM-SNEC we lost important feature of SOM - the ability of data visualization on a single map.

4 Conclusions

A new method SOM-MH for using SOM as a classifier was presented. It uses neuron's class membership and Metropolis-Hastings algorithm to control neuron's learning process. This can be interpreted as simulating neuron's hesitation during the learning or as simulated annealing of class membership. The hesitation of neuron decrease during the learning. The proposed method was compared to other state-of-art methods for using SOM in classification tasks. Test results confirm that the proposed method improve accuracy of classification. The other supervised clustering algorithms can be improved with proposed method. Matlab implementation of the SOM-MH model is available at `http://home.elka.pw.edu.pl/~pplonski/som_mh`.

References

1. Asuncion, A., Newman, D.J.: UCI Mmachine Learning Repository. University of California, Irvine, School of Information and Computer Sciences (2007)
2. Brereton, R.G.: Self Organising Maps for Visualising and Modelling. Chem. Cent. J. 6 (2012)
3. Dozono, H., Tokushima, H., Hara, S., Noguchi, Y.: An Algorithm of SOM using Simulated Annealing in the Batch Update Phase for Sequence Analysis. In: 5th Workshop on Self-Organizing Maps, pp. 171–178 (2005)
4. Fessant, F., Aknin, P., Oukhellou, L., Midenet, S.: Comparison of Supervised Self-Organizing Maps Using Euclidian or Mahalanobis Distance in Classification Context. In: Mira, J., Prieto, A.G. (eds.) IWANN 2001. LNCS, vol. 2084, pp. 637–644. Springer, Heidelberg (2001)
5. Fiannaca, A., Di Fatta, G., Gaglio, S., Rizzo, R., Urso, A.: Improved SOM Learning Using Simulated Annealing. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4668, pp. 279–288. Springer, Heidelberg (2007)
6. Hastings, W.K.: Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika 57, 97–109 (1970)
7. Hinton, G.E., Dayan, P., Revow, M.: Modeling the manifolds of Images of Handwritten Digits. IEEE Trans. Neural Netw. 8, 65–74 (1997)
8. Hu, W., Xie, D., Tan, T., Maybank, S.: Learning Activity Patterns Using Fuzzy Self-Organizing Neural Network. IEEE T. Syst. Man. Cy. B. 34, 1618–1626 (2004)
9. Kästner, M., Villmann, T.: Fuzzy Supervised Self-Organizing Map for Semi-supervised Vector Quantization. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS, vol. 7267, pp. 256–265. Springer, Heidelberg (2012)

10. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Sci. 220, 671–680 (1983)
11. Kohonen, T.: Self-organized Formation of Topologically Correct Feature Maps. Biol. Cybern. 43, 59–69 (1982)
12. Kohonen, T.: The Self-Organizing Map. Proc. IEEE 78, 1464–1480 (1990)
13. Kohonen, T., Oja, E., Simula, O., Visa, A., Kangas, J.: Engineering Applications of the Self-organizing Map. Proc. IEEE 84, 1358–1384 (2002)
14. Melssen, W., Wehrens, R., Buydens, L.: Supervised Kohonen Networks for Classification Problems. Cemometr. Intell. Lab. 83, 99–113 (2006)
15. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equations of State Calculations by Fast Computing Machines. J. Chem. Phys. 21, 1087–1092 (1953)
16. Midenet, S., Grumbach, A.: Learning Associations by Self-Organization: The LASSO model. Neurocomputing 6, 343–361 (1994)
17. Muruzabal, J.: On the Emulation of Kohonen's Self-Organization via Single-Map Metropolis-Hastings Algorithms. In: 9th International Conference on Conceptual Structures, pp. 346–355 (2001)
18. Osowski, S., Linh, T.H.: Fuzzy Clustering Neural Network for Classification of ECG Beats. In: International Joint Conference on Neural Networks, pp. 26–32. IEEE Press, New York (2000)
19. Płoński, P., Zaremba, K.: Improving Performance of Self-Organising Maps with Distance Metric Learning Method. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS, vol. 7267, pp. 169–177. Springer, Heidelberg (2012)
20. Sohn, S., Dagli, C.H.: Advantages of Using Fuzzy Class Memberships in Self-organizing Map and Support Vector Machines. In: International Joint Conference on Neural Networks, pp. 1886–1890. IEEE Press, New York (2001)
21. Song, X., Hopke, P.K.: Kohonen Neural Network as a Pattern Recognition Method Based on the Weight Interpretation. Anal. Chim. Acta. 334, 57–66 (1996)
22. Wongravee, K., Lloyd, G.R., Silwood, C.J., Grootveld, M., Brereton, R.G.: Supervised Self Organizing Maps for Classification and Determination of Potentially Discriminatory Variables: Illustrated by Application to Nuclear Magnetic Resonance Metabolomic Profiling. Anal. Chem. 82, 628–638 (2010)

Decoupled 2-D DOA Estimation Algorithm Based on Cross-Correlation Matrix for Coherently Distributed Source

Yinghua Han*, Jinkuan Wang, Qiang Zhao, and Peng Han

Northeastern University at Qinhuangdao, China
yhhan723@126.com

Abstract. A computationally efficient method for estimating two-dimensional (azimuth and elevation) direction-of-arrival (2-D DOA) of coherently distributed source is presented. Since the coherently distributed source is characterized by four parameters, the azimuth DOA, angular spread of the azimuth DOA, the elevation DOA, and angular spread of the elevation DOA, the computational complexity of the parameter estimation is normally highly demanding. A low-complexity estimation algorithm is proposed based on deduced Schur-Hadamard product steering vector which enables the estimation of 2-D DOA decoupled from that of angular spread of sources. The estimator constructs cross-correlation matrix from subarrays. And then the closed form solution of the elevation and azimuth DOA estimation can be obtained sequentially. Therefore, the proposed method avoids computationally demanding spectral search step and does not involve any eigen decomposition or singular value decomposition as in common subspace techniques such as MUSIC and ESPRIT. Numerical examples illustrate the performance of the method.

Keywords: Coherently distributed source, 2-D DOA estimation, Cross-correlation, Angular spread.

1 Introduction

Estimation of 2-D DOA is a key problem in array signal processing field such as radar, sonar, radio astronomy, and mobile communication systems[1-2]. Many signal source localization algorithm has focused on sources that are modeled as points in space. In point source model assumption, the source energy is concentrated at discrete angles that are referred to as the source DOA. However, in applications signal reflection and scattering phenomena at the source vicinity may result in angular spreading of the source energy, which degrade the performance of any array signal processing algorithm that uses a point source model. In this complex situation, a distributed source model will be more appropriate than the point source one [3-8].

Some typical estimators have been proposed for azimuth-only estimation of the DOA and angular spread of coherently or incoherently distributed source [5-8]. All these methods are involved joint spectral searching and computationally intensive.

* Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 157–165, 2012.

Some low-complexity estimators have also been given in [9-13]. In the low-complexity algorithms, some of them are sequential 1-D algorithms instead of joint 2-D searching [10]. Others are simpler but suboptimal solutions can be achieved by the subspace-based approach, which relies on signal subspace and noise subspace [11-13].

Recently, researchers have focuses on 2-D DOA estimation for distributed sources. However, for the problem of estimating the 2-D DOAs, the distributed source is characterized by four parameters, the azimuth DOA, angular spread of the azimuth DOA, the elevation DOA, and angular spread of the elevation DOA, the computational complexity of parameter estimation is normally highly demanding. Simpler but suboptimal solutions can be achieved by SOS algorithm [14], which relies on eigendecomposition and 1-D searching for estimating the azimuth and elevation DOA. Using a uniform circular array, 2-D DOA and angular spreads are estimated by a 2-D joint searching method in [15].

In this paper, we consider the coherently distributed source model and propose a low-complexity 2-D DOA estimation method using three uniform linear arrays. Based on the special array geometry, the cross-correlation matrix from signals received at subarrays is constructed. And then the elevation and azimuth DOA estimation can be obtained sequentially. The resultant decoupled algorithm avoids spectral search step and does not involve any eigendecomposition or singular value decomposition.

2 System Model

Consider an array configuration which consists of three uniform linear subarrays as in Fig.1. The interspacing d between sensors in each subarray is equal to a half-wavelength of incident signals. Let X , Z and W denotes the three subarrays and each linear array consists of M elements.

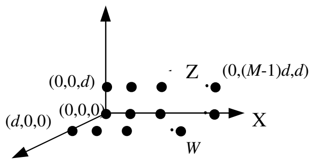


Fig. 1. The array configuration for 2-D DOA estimation of coherently distributed source

Suppose that there are q narrow-band sources impinging on the array. The received vector of subarray X can be written as

$$\boldsymbol{X}(t)=\sum_{i=1}^{q}\boldsymbol{S}_i(t)+\boldsymbol{N}_X(t) \tag{1}$$

where $\boldsymbol{X}(t)$ is the array snapshot vector, $\boldsymbol{S}_i(t)$ is the vector that describes the contribution of the ith source to the array output, and $\boldsymbol{N}_X(t)$ is additive zero-mean noise for subarray X uncorrelated from the signals.

In point source modeling, the baseband signal of the ith source is modeled as

$$\boldsymbol{S}_i(t) = s_i(t)\boldsymbol{a}(\theta_i,\phi_i) \tag{2}$$

where $s_i(t)$ is the complex envelope of the ith source, $\boldsymbol{a}(\theta_i,\phi_i) = \left[1 \quad \exp\left(-\mathrm{j}2\pi(d/\lambda)\sin\theta_i\sin\phi_i\right) \quad \cdots \quad \exp\left(-\mathrm{j}2\pi(M-1)(d/\lambda)\sin\theta_i\sin\phi_i\right)\right]^{\mathrm{T}}$ is the corresponding steering vector, θ_i and ϕ_i are the elevation and azimuth DOA, respectively, λ is the wavelength of the impinging signal.

In distributed source modeling, the source energy is considered to be spread over some angular volume. Hence, $\boldsymbol{S}_i(t)$ is written as

$$\boldsymbol{S}_i(t) = \iint \boldsymbol{a}(\vartheta,\varphi)\varsigma_i(\vartheta,\varphi,t)\mathrm{d}\vartheta\mathrm{d}\varphi \tag{3}$$

where $\varsigma_i(\vartheta,\varphi,t)$ is the angular signal density of the ith source and can be expressed as

$$\varsigma_i(\vartheta,\varphi,t) = s_i(t)\ell_i(\vartheta,\varphi;\boldsymbol{\mu}_i) \tag{4}$$

under the coherently distributed source assumptions. In (4), $\ell(\vartheta,\varphi;\boldsymbol{\mu})$ is a deterministic angular signal intensity function, and is parametrized by the vector $\boldsymbol{\mu} = (\theta,\sigma_\theta,\phi,\sigma_\phi)$ denoting the elevation DOA θ, angular spread σ_θ of the elevation DOA, the azimuth DOA ϕ, and angular spread σ_ϕ.

The steering vector of subarray X can be written as

$$\boldsymbol{b}_X(\boldsymbol{\mu}) = \iint \boldsymbol{a}(\vartheta,\varphi)\ell(\vartheta,\varphi;\boldsymbol{\mu})\mathrm{d}\vartheta\mathrm{d}\varphi \tag{5}$$

As a common example of the coherently distributed source, assume that the deterministic angular signal intensity function $\ell(\vartheta,\varphi;\boldsymbol{\mu})$ has the Gaussian shape as follows,

$$\ell(\vartheta,\varphi;\boldsymbol{\mu}) = \left(1\big/\left(2\pi\sigma_\theta\sigma_\phi\right)\right)\exp\left(-1/2\left((\vartheta-\theta)^2\big/\sigma_\theta^2 + (\varphi-\phi)^2\big/\sigma_\phi^2\right)\right) \tag{6}$$

The received signal vector in other subarrays Z and W can also be expressed as

$$\boldsymbol{Z}(t) = \sum_{i=1}^{q}\iint \boldsymbol{a}(\vartheta,\varphi)\ell(\vartheta,\varphi;\boldsymbol{\mu})\exp\left(-\mathrm{j}2\pi(d/\lambda)\cos\vartheta\right)s_i(t)\mathrm{d}\vartheta\mathrm{d}\varphi + \boldsymbol{N}_Z(t) \tag{7}$$

$$\boldsymbol{W}(t)=\sum_{i=1}^{q}\iint \boldsymbol{a}(\vartheta,\varphi)\ell(\vartheta,\varphi;\boldsymbol{\mu})\exp\left(-\mathrm{j}2\pi(d/\lambda)\sin\vartheta\cos\varphi\right)s_i(t)\,\mathrm{d}\vartheta\mathrm{d}\varphi+\boldsymbol{N}_W(t) \tag{8}$$

and the steering vectors are defined as $\boldsymbol{b}_Z(\boldsymbol{\mu})$ and $\boldsymbol{b}_W(\boldsymbol{\mu})$, respectively.

$$\boldsymbol{b}_Z(\boldsymbol{\mu})=\iint \boldsymbol{a}(\vartheta,\varphi)\exp\left(-\mathrm{j}2\pi(d/\lambda)\cos\vartheta\right)\ell(\vartheta,\varphi;\boldsymbol{\mu})\,\mathrm{d}\vartheta\mathrm{d}\varphi \tag{9}$$

$$\boldsymbol{b}_W(\boldsymbol{\mu})=\iint \boldsymbol{a}(\vartheta,\varphi)\exp\left(-\mathrm{j}2\pi(d/\lambda)\sin\vartheta\cos\varphi\right)\ell(\vartheta,\varphi;\boldsymbol{\mu})\,\mathrm{d}\vartheta\mathrm{d}\varphi \tag{10}$$

3 Decoupled 2-D DOA Estimation Algorithm Based on Cross-Correlation Matrix

In general, an optimum estimation method for point or distributed sources can provide an excellent performance at the cost of intensive computation. Since the computational complexity increases dramatically with high dimensional parameters, we have to sometimes find suboptimum methods to reduce the computational cost while sustaining the estimation performance within a tolerable level. It is noteworthy that a considerable simplification is possible by exploiting and utilizing the special array structure of the array geometry in the parameter estimation under coherently distributed source model also.

For Gaussian angular signal intensity, the steering vector $\boldsymbol{b}_X(\boldsymbol{\mu})$ can be written as

$$\begin{aligned}\left[\boldsymbol{b}_X(\boldsymbol{\mu})\right]_m=&\iint\exp\left(-\mathrm{j}2\pi(m-1)(d/\lambda)\sin\vartheta\sin\varphi\right)\left(1/\left(2\pi\sigma_\theta\sigma_\phi\right)\right)\times\\&\exp\left(-1/2\left((\vartheta-\theta)^2/\sigma_\theta^2+(\varphi-\phi)^2/\sigma_\phi^2\right)\right)\mathrm{d}\vartheta\mathrm{d}\varphi\end{aligned} \tag{11}$$

where $[\bullet]_m$ indicates the mth element of a vector. In distributed source, ϑ and φ are all around θ and ϕ. So with the change of variables $\vartheta-\theta=\tilde{\theta}$ and $\varphi-\phi=\tilde{\phi}$, $\tilde{\theta}$ and $\tilde{\phi}$ are small values. We can rewrite (11) as

$$\begin{aligned}\left[\boldsymbol{b}_X(\boldsymbol{\mu})\right]_m\approx&\iint\exp\left(-\mathrm{j}2\pi(m-1)(d/\lambda)\left(\sin\theta+\tilde{\theta}\cos\theta\right)\left(\sin\phi+\tilde{\phi}\cos\phi\right)\right)\times\\&\frac{1}{2\pi\sigma_\theta\sigma_\phi}\exp\left(-1/2\left(\tilde{\theta}^2/\sigma_\theta^2+\tilde{\phi}^2/\sigma_\phi^2\right)\right)\mathrm{d}\tilde{\theta}\mathrm{d}\tilde{\phi}\\=&\frac{1}{2\pi\sigma_\theta\sigma_\phi}\exp\left(-\mathrm{j}2\pi(m-1)(d/\lambda)\sin\theta\sin\phi\right)\times\\&\int\exp\left(-\mathrm{j}2\pi(m-1)(d/\lambda)\tilde{\phi}\sin\theta\cos\phi\right)\exp\left(-(1/2)\left(\tilde{\phi}^2/\sigma_\phi^2\right)\right)\mathrm{d}\tilde{\phi}\times\\&\int\exp\left(-\mathrm{j}2\pi(m-1)(d/\lambda)\tilde{\theta}\cos\theta\sin\phi\right)\exp\left(-(1/2)\left(\tilde{\theta}^2/\sigma_\theta^2\right)\right)\mathrm{d}\tilde{\theta}\end{aligned} \tag{12}$$

Let us consider the approximate form of $\boldsymbol{b}_X\left(\boldsymbol{\mu}\right)$. Using the integral formula [16],

$$\begin{aligned}&\int_{-\infty}^{\infty}\exp\left(-f^2x^2\right)\exp\left[\mathrm{j}p\left(x+\alpha\right)\right]\mathrm{d}x\\&=\sqrt{\pi}\exp\left(-p^2\big/\left(4f^2\right)\right)\exp\left(\mathrm{j}p\alpha\right)\big/f\end{aligned} \tag{13}$$

Equation (12) can be expressed as

$$\left[\boldsymbol{b}_X\left(\boldsymbol{\mu}\right)\right]_m\approx\exp\left(-\mathrm{j}2\pi\left(m-1\right)\left(d/\lambda\right)\sin\theta\sin\phi\right)\times\left[\boldsymbol{g}_1\right]_m\times\left[\boldsymbol{g}_2\right]_m \tag{14}$$

where $\left[\boldsymbol{g}_1\right]_m=\exp\left(-2\pi^2\left(m-1\right)^2\left(d/\lambda\right)^2\sin^2\theta\cos^2\phi\sigma_\phi^2\right)$, $\left[\boldsymbol{g}_2\right]_m=\exp\left(-2\pi^2\left(m-1\right)^2\left(d/\lambda\right)^2\cos^2\theta\sin^2\phi\sigma_\theta^2\right)$.

In the matrix form it can be extended to

$$\boldsymbol{b}_X\left(\boldsymbol{\mu}\right)=\boldsymbol{a}\left(\theta,\phi\right)\odot\boldsymbol{g}_1\odot\boldsymbol{g}_2 \tag{15}$$

where $\odot$ is the Schur-Hadamard or element product. $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are real-valued because of the symmetry assumption on angular signal intensity.

For the steering vector $\boldsymbol{b}_Z\left(\boldsymbol{\mu}\right)$, there is

$$\begin{aligned}\left[\boldsymbol{b}_Z\left(\boldsymbol{\mu}\right)\right]_m\approx&\iint\exp\left(-\mathrm{j}2\pi\left(m-1\right)\left(d/\lambda\right)\sin\left(\theta+\tilde{\theta}\right)\sin\left(\phi+\tilde{\phi}\right)\right)\times\\&\exp\left(-\mathrm{j}2\pi\left(d/\lambda\right)\left(\cos\theta-\tilde{\theta}\sin\theta\right)\right)\times\\&\frac{1}{2\pi\sigma_\theta\sigma_\phi}\exp\left(-1/2\left(\left(\vartheta-\theta\right)^2\big/\sigma_\theta^2+\left(\varphi-\phi\right)^2\big/\sigma_\phi^2\right)\right)\mathrm{d}\vartheta\mathrm{d}\varphi\end{aligned} \tag{16}$$

Using $2\pi\left(d/\lambda\right)\tilde{\theta}\approx0$, $\boldsymbol{b}_Z\left(\boldsymbol{\mu}\right)$ can be rewritten as

$$\begin{aligned}\left[\boldsymbol{b}_Z\left(\boldsymbol{\mu}\right)\right]_m\approx&\iint\exp\left(-\mathrm{j}2\pi\left(m-1\right)\left(d/\lambda\right)\sin\left(\theta+\tilde{\theta}\right)\sin\left(\phi+\tilde{\phi}\right)\right)\times\\&\exp\left(-\mathrm{j}2\pi\left(d/\lambda\right)\cos\theta\right)\times\\&\frac{1}{2\pi\sigma_\theta\sigma_\phi}\exp\left(-1/2\left(\left(\vartheta-\theta\right)^2\big/\sigma_\theta^2+\left(\varphi-\phi\right)^2\big/\sigma_\phi^2\right)\right)\mathrm{d}\vartheta\mathrm{d}\varphi\end{aligned} \tag{17}$$

According to (14) and (17), we can write the following equations,

$$\boldsymbol{b}_Z\left(\boldsymbol{\mu}\right)\approx\exp\left(-\mathrm{j}2\pi\left(d/\lambda\right)\cos\theta\right)\boldsymbol{b}_X\left(\boldsymbol{\mu}\right) \tag{18}$$

For the steering vector $\boldsymbol{b}_W\left(\boldsymbol{\mu}\right)$, we also have

$$\boldsymbol{b}_W\left(\boldsymbol{\mu}\right)\approx\exp\left(-\mathrm{j}2\pi\left(d/\lambda\right)\sin\theta\cos\phi\right)\boldsymbol{b}_X\left(\boldsymbol{\mu}\right) \tag{19}$$

To implement the proposed decoupled 2-D DOA estimation algorithm for a single coherently distributed source, we formulate a cross-correlation matrix between the signals received at the subarrays, i.e., we consider the following cross-correlation matrix,

$$\boldsymbol{R}_{XZ}^{k}=E\left\{\boldsymbol{X}^{k}\left(t\right)\left[\boldsymbol{Z}^{k}\left(t\right)\right]^{*}\right\}=\exp\left(\mathrm{j}2\pi\frac{d}{\lambda}\cos\theta\right)\times\left[\boldsymbol{g}_{1}\right]_{k}\times\left[\boldsymbol{g}_{2}\right]_{k}\times\left[\boldsymbol{g}_{1}\right]_{k}\times\left[\boldsymbol{g}_{2}\right]_{k}\times E\left\{s\left(t\right)\left[s\left(t\right)\right]^{*}\right\}+ E\left\{\boldsymbol{N}_{X^{k}}\left(t\right)\left[\boldsymbol{N}_{Z^{k}}\left(t\right)\right]^{*}\right\} \tag{20}$$

where $k\in\left[1,M\right]$, $\boldsymbol{X}^{k}\left(t\right)$ and $\boldsymbol{Z}^{k}\left(t\right)$ denote signals received at the *k*th sensors at subarrays *X* and *Z*, $\left[\ \right]^{*}$ is conjugate of a matrix. So the elevation DOA can easily be found as

$$\widehat{\theta}=\frac{1}{M}\sum_{k=1}^{M}\arccos\left(\left(\angle\boldsymbol{R}_{XZ}^{k}\right)/\left(2\pi\left(d/\lambda\right)\right)\right) \tag{21}$$

where $\angle\left(\bullet\right)$ stands for the phase angle of $\bullet$.

For estimating the azimuth DOA, the cross-correlation matrices are defined as follows,

$$\boldsymbol{R}_{XX}^{n}=E\left\{\boldsymbol{X}^{n+1}\left(t\right)\left[\boldsymbol{X}^{n}\left(t\right)\right]^{*}\right\}=\exp\left(\text{-j}2\pi\frac{d}{\lambda}\sin\theta\sin\phi\right)\times\left[\boldsymbol{g}_{1}\right]_{n+1}\times\left[\boldsymbol{g}_{2}\right]_{n+1}\times\left[\boldsymbol{g}_{1}\right]_{n}\times\left[\boldsymbol{g}_{2}\right]_{n}\times E\left\{s\left(t\right)\left[s\left(t\right)\right]^{*}\right\}+ E\left\{\boldsymbol{N}_{X^{n+1}}\left(t\right)\left[\boldsymbol{N}_{X^{n}}\left(t\right)\right]^{*}\right\} \tag{22}$$

and

$$\boldsymbol{R}_{XW}^{n}=E\left\{\boldsymbol{X}^{n}\left(t\right)\left[\boldsymbol{W}^{n}\left(t\right)\right]^{*}\right\}=\exp\left(\mathrm{j}2\pi\frac{d}{\lambda}\sin\theta\cos\phi\right)\times\left[\boldsymbol{g}_{1}\right]_{n}\times\left[\boldsymbol{g}_{2}\right]_{n}\times\left[\boldsymbol{g}_{1}\right]_{n}\times\left[\boldsymbol{g}_{2}\right]_{n}\times E\left\{s\left(t\right)\left[s\left(t\right)\right]^{*}\right\}+ E\left\{\boldsymbol{N}_{X^{n}}\left(t\right)\left[\boldsymbol{N}_{W^{n}}\left(t\right)\right]^{*}\right\} \tag{23}$$

where $n\in\left[1,M-1\right]$.

Equations (22) and (23) both include the azimuth DOA, so it can easily be found as

$$\widehat{\phi}=\frac{1}{M-1}\sum_{n=1}^{M-1}\arctan\left(\angle\left(-\boldsymbol{R}_{XX}^{n}\right)/\angle\left(\boldsymbol{R}_{XW}^{n}\right)\right) \tag{24}$$

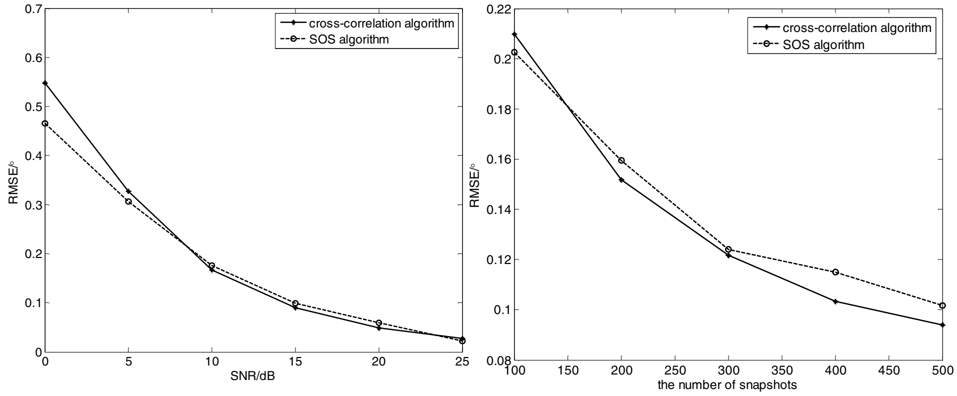
Equation (24) implies that the azimuth DOA $\widehat{\phi}$ can be estimated without any information of $\widehat{\theta}$, which avoids any error of $\widehat{\theta}$ effecting the estimation precision of $\widehat{\phi}$.

It's clear that the proposed method can estimate the 2-D DOA of coherently distributed source with the closed form solution in (21) and (24). It is a suboptimum algorithm and can be applicable when there exists a special array geometry. Regarding the major computational complexity, the proposed methods avoids computationally demanding spectral search step and does not involve any eigendecomposition or singular value decomposition as in common subspace techniques such as MUSIC and ESPRIT.

4 Simulation Results

Simulations of the 2-D DOA estimator proposed are completed to assess its performance. The elements of each antenna array are separated by a half-wavelength. The number of antenna elements in each axis is set to $M = 8$. We have considered a single narrowband coherently distributed source. The source has Gaussian shaped angular signal intensity functions with parameter $\boldsymbol{\mu}=\left[60^\circ\ ,\ 3^\circ\ ,\ 75^\circ\ ,\ 5^\circ\right]$.

The first simulation presented in Fig. 2(a) shows the root mean square error (RMSE) of the 2-D DOAs estimation in degrees for the proposed method compared with the SOS algorithm for different SNRs. The RMSE is defined as $\sqrt{E\left[\left(\hat{\theta}-\theta\right)^2+\left(\hat{\phi}-\phi\right)^2\right]}$. The received signals are obtained with 500 snapshots. The simulation results are computed over 500 trials. As it can be seen, the proposed algorithm has better estimation performance at high SNR. The explanation of this fact is that the bias of cross-correlation estimation has effected the 2-D DOA estimation in low SNR scenarios.



(a) (b)

Fig. 2. (a) RMSEs for 2-D DOA estimates versus SNR, (b) RMSEs for 2-D DOA estimates versus the number of snapshots

Second, let us consider the influence of the number of snapshots on the performance in Fig. 2(b) assuming that SNR=15dB. It is observed that the RMSEs in proposed method are rather small even when the number of snapshot is not large.

Third, Fig. 3 shows the RMSEs of the 2-D DOA estimates of the proposed and SOS methods as a function of angular spread. We have assumed that the number of snapshots $N = 500$ and $\mathrm{SNR} = 15\mathrm{dB}$. As angular spread increases from 1° to 5°, the performance of the proposed method degrades. However, it is clear that when the angular spread becomes large, the proposed algorithm can still give effective estimation.

The proposed method is a suboptimum method and applicable when there exists a special array geometry, while classical subspace algorithms can be used with arbitrary array geometry and generally provide almost optimum performance at the expense of higher computational complexity. So there is always a tradeoff between the performance and computational complexity.

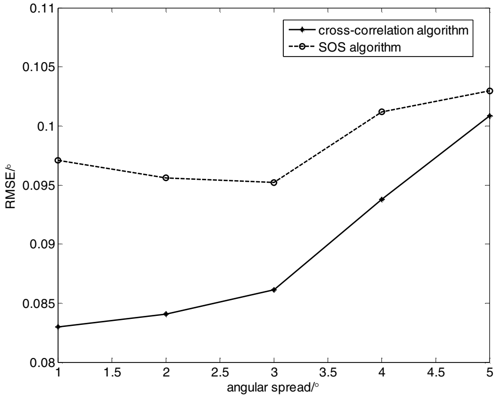


Fig. 3. RMSEs for 2-D DOA estimates versus angular spread

5 Conclusions

In this paper, we have considered the estimation of the parameters of a spatially distributed source with a view to provide a statistically and computationally efficient algorithm. Based on Schur-Hadamard product steering vector, the proposed algorithm is applied to a cross-correlation matrix constructed from the received signal at subarrays. And then the 2-D DOA can be estimated decoupledly. Unlike previous 2-D DOA estimators for distributed source, the proposed method is capable of estimating the azimuth and elevation angles without any peak-finding searching and eigenvalue decomposition.

Acknowledgements. This work was supported by the National Natural Science Foundation of China under Grant No. 60904035, 61004052 and 61104005, by Natural Science Foundation of Liaoning Province and Hebei Province under Grant No.201202073 and F2011501052, by The Central University Fundamental Research Foundation, under Grant. N110423004 and by Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20110042120015.

References

1. Ko, Y.H., Kim, Y.J., Yoo, H.I., Yang, W.Y., Cho, Y.S.: 2-D DOA Estimation with Cell Searching for a Mobile Relay Station with Uniform Circular Array. IEEE Trans. Communications 58, 2805–2809 (2010)

2. Gan, L., Gu, J., Wei, P.: Estimation of 2-D DOA for Noncircular Sources Using Simultaneous SVD Technique. IEEE Antenn. Wirel. Pr. 7, 385–388 (2008)
3. Monakov, A., Besson, O.: Direction Finding for an Extended Target with Possibly Non-symmetric Spatial Spectrum. IEEE Trans. Signal Proces. 52, 283–287 (2004)
4. Raich, R., Goldberg, J., Messor, H.: Bearing Estimation for a Distributed Source: Modeling, Inherent Accuracy Limitations and Algorithm. IEEE Trans. Signal Proces. 48, 429–441 (2000)
5. Hassanien, A., Shahbazpanahi, S., Gershman, A.B.: A Generalized Capon Estimator for Localization of Multiple Spread Sources. IEEE Trans. Signal Proces. 52, 280–283 (2004)
6. Lee, J., Joung, J., Kim, J.D.: A method for the Direction-of-Arrival Estimation of Incoherently Distributed Sources. IEEE Trans. Veh. Technol. 57, 2885–2893 (2008)
7. Xiong, Y., Zhang, G.Y., Tang, B., Cheng, H.: Blind Identification and DOA Estimation for Array Sources in Presence of Scattering. J. Syst. Eng. Electron. 22, 393–397 (2011)
8. Meng, Y., Stoica, P., Wong, K.M.: Estimation of the Direction of Arrival of Spatially Dispersed Signals in Array Processing. IEE P-Radar Son. Nav. 43, 1–9 (1996)
9. Shahbazpanahi, S., Valaee, S., Gershman, A.B.: A Covariance Fitting Approach to Parametric Localization of Multiple Incoherently Distributed Sources. IEEE Trans. Signal Proces. 52, 592–600 (2004)
10. Souden, M., Affes, S., Benesty, J.: A Two-Stage Approach to Estimate the Angles of Arrival and the Angular Spreads of Locally Scatters Sources. IEEE Trans. Signal Proces. 56, 1968–1983 (2008)
11. Zoubir, A., Wang, Y., Charge, P.: Spatially Distributed Sources Localization with a Subspace Based Estimator without Eigen Decomposition. In: Proceedings of ICASSP, pp. 1085–1088 (2007)
12. Zoubir, A., Wang, Y., Charge, P.: Efficient Subspace-Based Estimator for Localization of Multiple Incoherently Distributed Source. IEEE Trans. Signal Proces. 56, 532–542 (2008)
13. Shahbazpanahi, S., Valaee, S., Bastani, M.H.: Distributed Source Localization Using ESPRIT Algorithm. IEEE Trans. Signal Proces. 49, 2169–2178 (2001)
14. Lee, J., Song, L., Kwon, H., Lee, S.R.: Low-Complexity Estimation of 2D DOA for Coherently Distributed Sources. Signal Proces. 83, 1789–1802 (2003)
15. Wan, Q., Peng, Y.N.: Low-Complexity Estimator for Four-Dimensional Parameters under a Reparameterised Distributed Source Model. IEE Proc.-Radar. Sonar Nav. 148, 313–317 (2001)
16. Gradshteyn, I.S., Ryzhik, I.M.: Table of Integrals, Series, and Products. Academic Press, Orlando (1980)

Feature Extraction by Nonnegative Tucker Decomposition from EEG Data Including Testing and Training Observations

Fengyu Cong[1,*], Anh Huy Phan[2], Qibin Zhao[2], Qiang Wu[3],
Tapani Ristaniemi[1], and Andrzej Cichocki[2]

[1] Department of Mathematical Information Technology, University of Jyväskylä, Finland
[2] Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Japan
[3] School of Information Science and Engineering, Shandong University, Shandong, China
{fengyu.cong,tapani.ristaniemi}@jyu.fi,
{phan,qbzhao,cia}@brain.riken.jp, wuqiang@sdu.edu.cn

Abstract. The under-sample classification problem is discussed for 21 normal childrenand 21 children with reading disability. We first rejected data of one subject in each group and produced 441 sub-datasets including 40 subjects in each. Regarding each sub-dataset, we extracted features through nonnegative Tucker decomposition (NTD) from event-related potentials, and used the leave-one-out paradigm for classification. Averaged accuracies over 441 sub-datasets were 77.98% (linear discriminate analysis), 73.55% (support vector machine), and 76.97% (adaptive boosting). In summary, assuming K observations with known labels, for the new observation without the group information, the feature of the new observation can be extracted through performing NTD to extract features from data of all observations (K+1). Since the group information of the first K observations is known, their features can train the classifier, and then, the trained classifier recognizes new features to determine the group information of new observation.

Keywords: Classification, Event-related potential, Mismatch negativity, Multi-domain feature extraction, Nonnegative Tucker decomposition, Undersample.

1 Introduction

Brain-Computer interface (BCI) has become a bridge to connect brain states and external devices. Brain states can be often represented by electroencephalography (EEG) in BCI [29]. Indeed, EEG data can be divided into three groups including spontaneous EEG [24], event-related potentials (ERPs) [16] and ongoing EEG [10]. Different kinds of EEG data have very different properties. For the spontaneous EEG, the data are usually collected in the rest-state of the participant, and the oscillations are often analyzed [24]. Regarding ERPs, a short external stimulus is repeated hundreds of times; each time can be called as each single trial or epoch; EEG data of those single trials are usually averaged to produce ERPs; the peak amplitudes and latencies of ERPs are frequently analyzed [16]. Ongoing EEG are often recorded in

[*] Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 166–173, 2012.

the circumstance of real experience of a participant, for example, listening to natural, long and continuous music; oscillations can be analyzed; and the temporal course of an oscillation can be linked to the stimulus [10].

For EEG based BCI studies, the event-related designs have been extensively investigated [29]. The single-trial EEG data are mostly analyzed in BCI instead of the averaged EEG used in the conventional ERP analysis [29]. In BCI, classifying brain states based on machine learning methods [1] is one of the core steps [29]. For such classification, the number of single trials is the number of observations and the number of features extracted from single-trial EEG data is the number of variables. Since the number of analyzed single trials can be hundreds in BCI [32-33], the machine learning based classification works very well. Indeed, for the single-trial ERP data in BCI, the basic assumption is that the elicited ERP activity in each of analyzed single trials is observable [29], [32]. However, some small ERPs tend to be invisible in the single trial ERP data and averaging over hundreds of single trials is usually required to produce the small ERPs [20-21], [23]. From this point of view, classification in BCI cannot be performed on single trials data of such small ERPs.

Small ERPs can be very important for cognitive research. For example, mismatch negativity (MMN) can reflect a subject's ability of passively detecting the deviant among the repeated stimuli in a regular auditory pattern [20], and can be applied to study the automatic auditory brain functions, related to discrimination and perception, of the brain of children with delayed language development [15], [17]. Specifically, its peak amplitude has been acknowledged to be an endophenotype to study normal children and children with disorders, owing to the smaller peak amplitude of MMN generated by the latter group [11] and the different topographies of peak amplitudes between the normal children and children with reading disability (RD) [13]. Because MMN is very small the group-level analysis is still the main methodology for its analysis [22]. Usually, statistical tests are performed on the peak measurements of participants of different groups, and the significance of the difference between/among groups is reported [16]. Indeed, such studies are targeted to search the difference between/among groups of participants from the view of data processing, which is very similar for the machine learning based classification when a participant is regarded as one observation. Compared with statistical tests, the machine learning based classification is much more powerful to discriminate different observations [1]. Hence, it can be significant to implement the machine learning based classification on the ERP data that are the averaged EEG, which is towards to a different BCI in contrast to the single-trial ERP data.

In our previous study [8], we performed the classification on the features extracted from the ERP data which were the averages over single trials. For such applications, the number of participants is the number of observations for the machine learning based classification. However, the problem is that the number of such observations in ERP studies is usually limited to be dozens [16], and then, the classification belongs to the under-sample problem. Indeed, regarding such a problem, the tensor discriminate analysis (TDA) has been proposed [30], [31]. For TDA, features are extracted through a supervision manner. If TDA is applied for ERP studies, one drawback is that the features are hardly interpreted by the physiological properties of ERPs. In this study, we will design a new paradigm for under-sample classification based on the multi-domain features of MMN [7], [8] extracted by nonnegative Tucker decomposition (NTD) [25], [26].

2 Method

2.1 Data Description

The data were collected in University of Jyväskylä, Finland. In this study, the data of 21 normal children and 21 children with RD were taken for analysis. For detailed information on how to categorize the children, please refer to Huttunen et al., [13]. The CONT group consisted of 11 boys and 10 girls and the mean age of this group was 11 years 6 months (age range: 8 years 8 months to 13 years 2 months); and the RD group included 16 boys and 5 girls, and their mean age was 11 years 9 months (age range: 8 years 8 months to 14 years 2 months). In the study, an uninterrupted sound under the oddball paradigm was used to elicit MMN. This paradigm consisted of the uninterrupted sound, alternating 100ms sine tones of 600Hz and 800Hz (repeated stimuli, see Fig.1). There was no pause between the alternating tones and their amplitudes did not change. During the experiment 15% of the 600Hz tones were randomly replaced by shorter ones of 50ms and 30ms duration (called as dev50 and dev30 hereinafter). The deviants consisted of 7.5% of 50ms deviants and 7.5% 30ms deviants. There were at least six repetitions of the alternating 100ms tones between two deviants.

EEG recordings at nine channels (frontal F3, Fz, F4; central C3, Cz, C4; parietal Pz and mastoids M1, M2) were collected with Electro-Cap International 20-electrode cap using the standard 10-20 system. The potentials were referenced to the tip of nose. After a band-pass filter of 0.1–30Hz was applied, EEG was downsampled with the rate of 200Hz. Recording started 300ms before the onset of a deviant stimulus and lasted 350ms after the onset of a deviant. Thus each trial contained the recordings of 650 ms, i.e., 130 samples. In order to remove artifacts, two types of exclusion criteria were applied. Firstly, trials with amplitude exceeding ±100 μV were rejected. Secondly, trials with recordings of zero variance were deleted. After the artifacts rejection, the mean number of trials per child was about 331 with the standard deviation of 21.6. In order to obtain the stable MMN, the kept trials were averaged for each subject, followed by removing the baseline formed by the average of the first 300 ms recordings. Then, the Morlet wavelet transform was performed on the average to obtain the time-frequency representation (TFR) of MMN. For the wavelet, the half wavelet length was set to be six for the optimal resolutions of the frequency and time [28]; the frequency range was set from 2 to 8.5Hz, and this was because the optimal frequency band of MMN in our dataset was in this range[3], [14]; 256 frequency bins were logarithmically distributed within this frequency range.

Then, the data was well prepared for the following feature extraction by NTD. It should be noted that only the data at one electrode under dev50 was chosen for analysis in this study, and the task was to classify the normal children and children with RD.

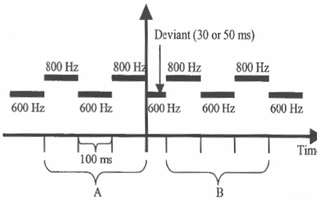


Fig. 1. Stimuli sequence

2.2 Nonnegative Tucker Decomposition

For a given *N*th-order tensor$\underline{\mathbf{Y}} \in \Re_{+}^{I_1 \times I_2 \times \cdots \times I_N}$**,** NTD model [25], [26] usually reads,

$$\underline{\mathbf{Y}} \approx \underline{\mathbf{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}, \tag{1}$$

where, $\mathbf{A}^{(n)} = \left[\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \cdots, \mathbf{a}_{J_n}^{(n)}\right] \in \Re^{I_n \times J_n}, (n = 1,2,\cdots,N)$are common factors or loadings represented by component matrices, and $\underline{\mathbf{G}} \in \Re^{J_1 \times J_2 \times \cdots \times J_N}$ is core tensor. Practical NTD algorithms are usually in terms of alternative least squares (ALS) minimization of the squared Euclidean distance (Frobenius norm) subject to the nonnegativity constraints [27]. An alternative approach for NTD is the use of the all-at-once algorithms which simultaneously update all the factors and the core tensor such as the damped Gauss-Newton algorithm [27].

For feature extraction [25], [26], the last mode, i.e., the mode-*N*, is usually the sample/instance and $\underline{\mathbf{G}}$ represents the multi-domain features. Subsequently, the corresponding learning processing regarding NTD model is as the following

$$D_F\left(\underline{\mathbf{Y}} \middle| \underline{\mathbf{G}}, \{\mathbf{A}\}\right) = \frac{1}{2}\left\|\underline{\mathbf{Y}} - \underline{\mathbf{G}} \times_1 \mathbf{A}^{(1)} \cdots \times_{N-1} \mathbf{A}^{(N-1)}\right\|_F^2 = \frac{1}{2}\left\|\underline{\mathbf{Y}} - \underline{\mathbf{G}} \times_{-(N)} \{\mathbf{A}\}\right\|_F^2. \tag{2}$$

In a compact form of tensor products [20], the multiplicative update rule for factors $\mathbf{A}^{(n)}$ $(n = 1,2,\cdots,N-1)$ used in this study is given by

$$\mathbf{A}^{(n)} \leftarrow \mathbf{A}^{(n)} \circledast \langle \underline{\mathbf{Y}} \times_{-(n,N)} \{\mathbf{A}^{\mathrm{T}}\}, \underline{\mathbf{G}} \rangle_{-n} \oslash \left(\mathbf{A}^{(n)} \langle \underline{\mathbf{G}}, \underline{\mathbf{G}} \times_{-(n,N)} \{\mathbf{A}^{\mathrm{T}}\mathbf{A}\} \rangle_{-n}\right) \tag{3}$$

and the update rule [20] for the core tensor $\underline{\mathbf{G}}$ used in this study is as

$$\underline{\mathbf{G}} \leftarrow \underline{\mathbf{G}} \circledast \left(\underline{\mathbf{Y}} \times_{-(N)} \{\mathbf{A}^{\mathrm{T}}\}\right) \oslash \left(\underline{\mathbf{G}} \times_{-(N)} \{\mathbf{A}^{\mathrm{T}}\mathbf{A}\}\right) \tag{4}$$

where $\circledast$ and $\oslash$ denote the Hadamard product and division.

2.3 Feature Extraction and Classification Paradigm

In this study, each group had 21 subjects. From the view of the leave-one-out policy, data of 41 subjects can be used to extract the training features by NTD and learn the feature space, and then, the left data can be projected to the learned feature space to produce the testing features by NTD. We did not adopt such conventional machine learning based classification paradigm in this study because the number of the observations in each group was not enough for learning the feature space.

Considering the undersample problem, we design a new paradigm for the classification. It includes two parts. One is to validate the separability of the two groups through the features extracted by NTD, and the other is to test the paradigm. We take the example of the data in this study to illustrate this new paradigm.

Firstly, we can learn the feature space and extract features through performing NTD on data of all 42 children. Indeed, the feature extraction is not supervised and the group information is not exploited here. Then, we try to classify the two groups with the derived features along the leave-one-out policy. We use features of 41 subjects to train the classifier and test the feature of the left one. The feature of every subject is tested. By this way, we can determine whether the two groups are able to be classified or not. If they can be classified, the paradigm is useful. Otherwise it is no use. This is the first step.

Secondly, data of one subject in each group is rejected, and then, data of 40 subjects are left for the feature extraction and classification. We use features of 39 subjects to train the classifier and test the feature of the left one. The feature of every subject is tested. In case each group has 21 observations, there are 441 different sub-datasets including 40 observations and any two sub-datasets at least has one different observation. Finally, the accuracies for classification in 441 sub-datasets represent the ability of the proposed paradigm to classify two groups with the features extracted by NTD.

We used three classifiers including the linear discriminate analysis (LDA)[12], support vector machine (SVM)[12], and adaptive boosting (Adaboost) [12]. For SVM, the 'RBF' nonlinear kernel was trained.

2.4 Data Processing

The features were extracted by NTD from EEG data at F3. This is because the normal children have larger magnitude of MMN peak in the right hemisphere, and the children with RD have larger one in the left hemisphere [13]. Thus, the data at F3 should be more discriminative between two groups in contrast to the data at other electrodes. As mentioned above, for the first step of the proposed method, 42 subjects' data composed a third-order tensor including the dimensions of frequency by time by subject; regarding the second step, 40 subjects' data were used. For the spectral and temporal factors, five and two components were extracted by NTD, respectively. Hence, ten features were extracted by NTD. And then, two most discriminative features were chosen for the classification. These parameters were chosen based on cross-validation.

3 Results

For the first step using all the data including 42 subjects, the accuracies of the classification of normal children and children with RD were 0.8095 (LDA), 0.8095 (SVM), and 0.9048 (Adaboost). Regarding the second step, the averaged accuracy over 441 sub-datasets including 40 subjects in each were 0.7798 (LDA), 0.7355 (SVM), and 0.7697 (Adaboost). Fig.2 shows the accuracy of the classification in every sub-dataset. The horizontal ordinate represents the accuracy of the classification, and the vertical ordinate denotes the number of datasets at certain accuracy.

Due to the limitation of the space of this study, the waveforms of ERPs and the single-trial ERP data are not shown. Please refer to [4], [9] for more details.

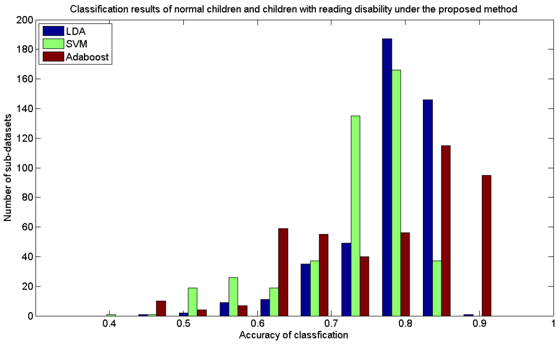


Fig. 2. Accuracies of classification in sub-datasets

4 Conclusion and Discussion

In this study, we propose a new classification paradigm that the features are extracted by NTD from data including the testing and the training observations. We show that such features can be used for the further classification when the undersample problem appears in BCI regarding ERPs. Due to the space limitation, we did not compare the proposed algorithm with other undersample problem solutions in this study.

Indeed, for NTD, the conventional machine learning based classification first learns the feature space through performing NTD on the training dataset, and then projects the testing data onto the learned feature space to produce the testing features [8]. Such a paradigm should work well in case the number of observations for the classification is enough to learn the feature space. However, in the study of ERPs, when a subject is regarded as an observation, the undersample problem often happens due to the limited number of observations. Hence, the conventional machine learning based feature extraction and classification should be modified to resolve this problem. The supervised feature extraction through tensor has been proposed [31]. However, since it is difficult to interpret the features extracted by such supervised learning from the psychological knowledge of an ERP, we do not exploit such methods in our study.

In this study, firstly, we learn the feature space and extract features of all observations together. Such a procedure is not supervised since we do not exploit the group information. Nevertheless, it is different from the conventional machine learning based feature extraction mentioned above. This results in that the accuracy of the classification using the features extracted from all data is just the special case. However, such obtained accuracy can illustrate whether the two or more groups can be classified or not through the extracted features with a classifier.

Secondly, in order to test the idea extracting features from data including the testing and the training observations, it is reasonable to reject some data from each group to form a sub-dataset, and to repeat the same procedure in the first step. After all possibilities to forming the sub-datasets are tested, the accuracy of the classification can be reliable.

Results showed that the accuracy of the classification of normal children and children with RD was around 80% in most sub-datasets when LDA was used, indicating the success of the proposed feature extraction and classification paradigm.

In summary, assuming we have K observations including two groups, when the new observation without knowing the group information comes, the feature of the new observation can be obtained through performing NTD to extract features from data of all observations (K+1). Since the group information of the first K observations is known, their features can train the classifier, and then, the trained classifier recognizes the new feature to determine the group information of the new observation.

It should be noted that group-level analysis using ERP is very important in the cognitive research as mentioned earlier. In order to facilitate such analysis, it is necessary to formulate the tensor including data of different subjects. This is because the component extracted by NTD has the inherent variance indeterminacy [2] which cannot be corrected. This is different from independent component analysis (ICA). Although the variance of a component extracted by ICA is not determined either, the back projection of the component to the sensor field can correct the indeterminacy when the decomposition of ICA is satisfactory in practice [5], [6], [18], [19]. From this point of view, ICA can be performed on the individual dataset for group-level analysis of ERPs [4], [9], but NTD has to be applied on the datasets including groups of subjects [7], [8]. Hence, developing fast NTD algorithm is critical and significant in the study of ERPs.

Acknowledgments. CF thanks sponsorship from Research and Innovation Office of University of Jyväskylä for international mobility grants (2009, 2010). The authors thank Professor Heikki Lyytinen and Dr. Tiina Huttunen-Scott for ERP studies.

References

1. Bishop, C.M.: Pattern recognition and machine learning, 1st edn. Springer, Singapore (2006)
2. Cichocki, A., Zdunek, R., Phan, A.H., et al.: Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis. John Wiley (2009)
3. Cong, F., Huang, Y., Kalyakin, I., et al.: Frequency Response Based Wavelet Decomposition to Extract Children's Mismatch Negativity Elicited by Uninterrupted Sound. J. Med. Biol. Eng. 32, 205–214 (2012)
4. Cong, F., Kalyakin, I., Li, H., et al.: Answering Six Questions in Extracting Children's Mismatch Negativity through Combining Wavelet Decomposition and Independent Component Analysis. Cogn. Neurodynamics 5, 343–359 (2011)
5. Cong, F., Kalyakin, I., Ristaniemi, T.: Can Back-Projection Fully Resolve Polarity Indeterminacy of ICA in Study of ERP? Biomed. Signal Process. 6, 422–426 (2011)
6. Cong, F., Kalyakin, I., Zheng, C., et al.: Analysis on Subtracting Projection of Extracted Independent Components from EEG Recordings. Biomed. Tech. 56, 223–234 (2011)
7. Cong, F., Phan, A.H., Astikainen, P., Zhao, Q., Hietanen, J.K., Ristaniemi, T., Cichocki, A.: Multi-domain Feature of Event-Related Potential Extracted by Nonnegative Tensor Factorization: 5 vs. 14 Electrodes EEG Data. In: Theis, F., Cichocki, A., Yeredor, A., Zibulevsky, M. (eds.) LVA/ICA 2012. LNCS, vol. 7191, pp. 502–510. Springer, Heidelberg (2012)
8. Cong, F., Phan, A.H., Lyytinen, H., Ristaniemi, T., Cichocki, A.: Classifying Healthy Children and Children with Attention Deficit through Features Derived from Sparse and Nonnegative Tensor Factorization Using Event-Related Potential. In: Vigneron, V., Zarzoso, V., Moreau, E., Gribonval, R., Vincent, E. (eds.) LVA/ICA 2010. LNCS, vol. 6365, pp. 620–628. Springer, Heidelberg (2010)
9. Cong, F., Kalyakin, I., Huttunen-Scott, T., et al.: Single-Trial Based Independent Component Analysis on Mismatch Negativity in Children. Int. J. Neural Syst. 20, 279–292 (2010)
10. Cong, F., Phan, A.H., Zhao, Q., Nandi, A.K., Alluri, V., Toiviainen, P., Poikonen, H., Huotilainen, M., Cichocki, A., Ristaniemi, T.: Analysis of Ongoing EEG Elicited by Natural Music Stimuli Using Nonnegative Tensor Factorization. In: Proceeding of The 2012 European Signal Processing Conference (EUSIPCO 2012), Bucharest, Romania, August 27-31, pp. 494–498 (2012)
11. Duncan, C.C., Barry, R.J., Connolly, J.F., et al.: Event-Related Potentials in Clinical Research: Guidelines for Eliciting, Recording, and Quantifying Mismatch Negativity, P300, and N400. Clin. Neurophysiol. 120, 1883–1908 (2009)
12. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting. J. Comput. Syst. Sci. 55, 119–139 (1997)
13. Huttunen, T., Halonen, A., Kaartinen, J., et al.: Does Mismatch Negativity show Differences in Reading-Disabled Children Compared to Normal Children and Children with Attention Deficit? Dev. Neuropsychol. 31, 453–470 (2007)

14. Kalyakin, I., Gonzalez, N., Joutsensalo, J., et al.: Optimal Digital Filtering Versus Difference Waves on the Mismatch Negativity in an Uninterrupted Sound Paradigm. Dev. Neuropsychol. 31, 429–452 (2007)
15. Leppanen, P.H., Lyytinen, H.: Auditory Event-Related Potentials in the Study of Developmental Language-Related Disorders. Audiol. Neurootol. 2, 308–340 (1997)
16. Luck, S.J.: An Introduction to the Event-Related Potential Technique. The MIT Press, Cambridge (2005)
17. Lyytinen, H., Guttorm, T.K., Huttunen, T., et al.: Psychophysiology of Developmental Dyslexia: A Review of Findings Including Studies of Children at Risk for Dyslexia. J. Neurolinguist. 18, 167–195 (2005)
18. Makeig, S., Jung, T.P., Bell, A.J., et al.: Blind Separation of Auditory Event-Related Brain Responses into Independent Components. Proc. Natl. Acad. Sci. U.S.A. 94, 10979–10984 (1997)
19. Makeig, S., Westerfield, M., Jung, T.P., et al.: Functionally Independent Components of the Late Positive Event-Related Potential during Visual Spatial Attention. J. Neurosci. 19, 2665–2680 (1999)
20. Näätänen, R.: Attention and Brain Functions. Lawrence Erlbaum Associates, Hillsdale (1992)
21. Näätänen, R., Gaillard, A.W., Mantysalo, S.: Early Selective-Attention Effect on Evoked Potential Reinterpreted. Acta. Psychol. (Amst) 42, 313–329 (1978)
22. Näätänen, R., Kujala, T., Kreegipuu, K., et al.: The Mismatch Negativity: An Index of Cognitive Decline in Neuropsychiatric and Neurological Diseases and in Ageing. Brain 134, 3432–3450 (2011)
23. Näätänen, R., Kujala, T., Winkler, I.: Auditory Processing that Leads to Conscious Perception: A Unique Window to Central Auditory Processing Opened by the Mismatch Negativity and Related Responses. Psychophysiology 48, 4–22 (2011)
24. Niedermeyer, E., Lopes da Silva, F.: Electroencephalography: Basic Principles, Clinical Applications, and Related Fields. Williams & Wilkins, Baltimore (2004)
25. Phan, A.H., Cichocki, A.: Extended HALS Algorithm for Nonnegative Tucker Decomposition and its Applications for Multiway Analysis and Classification. Neurocomputing 74, 1956–1969 (2011)
26. Phan, A.H., Cichocki, A.: Tensor Decomposition for Feature Extraction and Classification Problem. IEICE T, Fund. Electr. 1, 37–68 (2010)
27. Phan, A.H., Tichavsky, P., Cichocki, A.: Damped Gauss-Newton Algorithm for Nonnegative Tucker Decomposition, pp. 665–668 (2011)
28. Tallon-Baudry, C., Bertrand, O., Delpuech, C., et al.: Stimulus Specificity of Phase-Locked and Non-Phase-Locked 40 Hz Visual Responses in Human. J. Neurosci. 16, 4240–4249 (1996)
29. Tan, D.S., Nijholt, A.: Brain-Computer Interfaces: Applying our Minds to Human-Computer Interaction. In: Anonymous, p. 277. Springer, London (2010)
30. Tao, D.C., Li, X.L., Wu, X.D., et al.: Supervised Tensor Learning. Knowl. Inf. Syst. 13, 1–42 (2007)
31. Tao, D.C., Li, X.L., Wu, X.D., et al.: General Tensor Discriminant Analysis and Gabor Features for Gait Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 29, 1700–1715 (2007)
32. Zhang, Y., Zhao, Q., Jin, J., et al.: A Novel BCI Based on ERP Components Sensitive to Configural Processing of Human Faces. J. Neural Eng. 9, 026018 (2012)
33. Zhao, Q., Rutkowski, T.M., Zhang, L., et al.: Generalized Optimal Spatial Filtering using a Kernel Approach with Application to EEG Classification. Cogn. Neurodyn. 4, 355–358 (2010)

A New Approach for a Priori Client Threshold Estimation in Biometric Signature Recognition Based on Multiple Linear Regression

Arancha Simon-Hurtado*, Esperanza Manso-Martínez,
Carlos Vivaracho-Pascual, and Juan M. Pascual-Gaspar

Dep. Informática, University of Valladolid, Valladolid, Spain
{arancha,manso,cevp}@infor.uva.es,{pascualgaspar}@gmail.com

Abstract. This paper presents a novel approach to estimate (predict) the a priori client decision threshold for biometric recognition systems based on multiple linear regression. Biometric recognition is a complex classification problem where the goal is to classify a pattern (biometric sample) as belonging or not to a certain class (client). As in other pattern recognition problems, a correct estimation of the decision threshold is essential for optimizing the biometric system's performance. Our proposal is tested in biometric signature recognition, estimating thresholds for different system working points. A theoretical and practical performance analysis is presented, including a comparison with the state of the art, showing the advantages, in system performance, of our proposal.

Keywords: Threshold prediction, biometric signature recognition, multiple linear regression.

1 Introduction

This work focuses on the a priori client decision threshold estimation in biometric person recognition systems, where unique human characteristics (biometrics, e.g., iris, fingerprint, etc.) are used to recognize the user (client or Target Class, TC). Biometric recognition task can be split into two groups: identification (who is the owner of this biometric?) and verification or authentication (Am I the person I claim to be?). This second task is the one approached in this work.

Among the several biometrics used, signature presents some advantages [5], such as, for example, that it is widely accepted and commonly used in legal and commercial transactions as an authentication method. In addition, it is the second most important [10] of the behavioral biometrics. Signature verification can be split into: i) *Static or off-line*, where the signature written on paper is digitized, and ii) *Dynamic or on-line*, where users write their signature in a digitizing device. Static systems are restricted to use in legal cases. The experiments have been performed using dynamic signature recognition. Depending on the test conditions, two types of forgeries can be established: i) *skilled forgery*,

* Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 174–182, 2012.

where the impostor imitates the client signature, and ii) *random forgery*, where the impostor uses his/her own signature as a forgery.

In development tasks, the decision threshold can be estimated a posteriori. However, in real applications, the threshold must be established a priori, and it is fundamental to optimize the system performance: A good system operating with a wrong threshold becomes useless. Several important problems arise concerning threshold accuracy estimation in real world biometric applications: i) It is common to have only a few data from the TC, biasing the statistics estimation[12]. ii) It is difficult, or even impossible, to get an adequate Non-Target Class, NTC, (impostors) representation in some biometrics (e.g., it is not legal to get forgeries in manuscript signatures). iii) Generally, to simplify, the samples are supposed to be i.i.d., but in biometry this is not true; it is the so-called "biometric menagerie" that was first pointed out in [3] for speaker recognition, but the same phenomenon has been independently observed in other biometrics.

Here, a new approach to deal with these problems in a priori client threshold prediction is shown. Our proposal is based on the use of the Multiple Linear Regression (MLR). This proposal has been tested for multi-working points, i.e., it has been tried to predict thresholds for different, and representatives, system performance points (Sec. 4.2). Besides, a comparison with other prediction proposals in the literature has been performed, showing the advantages, in prediction accuracy, of our proposal.

The paper is organized as follows. We will begin (Sec. 2) with a brief analysis of the state of the art of threshold estimation in biometric recognition. The main characteristics of our proposal are shown in section 3, including a theoretical introduction to MLR. The experimental setup can be seen in section 4. The performance of our proposal is presented in section 5. The conclusions can be seen in section 6.

2 Related Works

Given a test sample X, the problem of biometric verification can be stated as a basic statistical hypothesis test:

$$H_0 : X \text{ is from client } C \quad \text{and} \quad H_1 : X \text{ is not from client } C$$

The decision between the two hypotheses is performed as shown in Eq. 1

$$P(X/H_0) \begin{cases} > \theta_C \ \textit{Accept } H_0 \\ < \theta_C \ \textit{Reject } H_0 \end{cases} \tag{1}$$

Where, θ_C is the client decision threshold, different for each client. Biometric verification is a pattern recognition problem, where each client C is represented by means of a model (HMM, GMM, ANN, etc.) λ^C. Then, $P(X/H_0)$, whose calculation is not a straightforward task, is estimated (approximated) by means of the classifier output (score) $s(X/\lambda^C)$. As a result, the decision for an authentication system based on user modeling is given by equation 2.

$$s(X/\lambda^C) \begin{cases} > \theta_C \; Accept \; H_0 \\ < \theta_C \; Reject \; H_0 \end{cases} \tag{2}$$

Not much works can be found focused on a priori client threshold estimation, $\hat{\theta}_C$. Perhaps one of the first attempt can be seen in [7], where the importance of setting thresholds in advance in practical situations is emphasized. In this work, Furui proposes the equation 3, based on empirical results. Later proposals are, in general, modifications of the Furui equation. In [12], a comparison among several methods to predict the client threshold is performed, achieving the best results with the proposal shown in Eq. 4. Similar approaches can be found in more recent works, as for example in [2], where equation 5 is proposed, or in [14], that proposes the equation 6. In [8], the same authors propose an actualization of the threshold with the data of the client collected during operation.

$$\hat{\theta}_C = \alpha(\hat{\mu}_C^N - \hat{\sigma}_C^N) + \beta \tag{3}$$

$$\hat{\theta}_C = \beta\hat{\mu}_C^N + (1-\beta)\hat{\mu}_C^M \tag{4}$$

$$\hat{\theta}_C = \alpha(\hat{\mu}_C^N + \beta\hat{\sigma}_C^N) + (1-\alpha)\hat{\mu}_C^M \tag{5}$$

$$\hat{\theta}_C = \hat{\mu}_C^M - \alpha\hat{\sigma}_C^M \tag{6}$$

In these equations, $\hat{\mu}_C^N$ and $\hat{\sigma}_C^N$ are the mean and standard deviation of the Non-Match (scores for NTC samples) distribution for the client C estimated by means of the so called *Cohort Gallery* (Sec. 4.1), $\hat{\mu}_C^M$ and $\hat{\sigma}_C^M$ are the mean and standard deviation of the Match (scores for TC samples) distribution for the client C, estimated by means of the so called *Client Gallery* (Sec. 4.1), and α and β are constant parameters which are set experimentally.

Our contribution is different in several ways. First, and important, a well founded theoretical technique is proposed, given us statistical tools for his evaluation and optimization. A pool of independent variables, besides of the mean and variance, has been proposed for a better modeling of the Match and Non-Match distributions (Sec. 3). The goal is to include in the model parameters not only associated with the assumption that the score distributions are gaussian. Following that shown in [15], some of the independent variables have been included to try to model the tail of the distributions, which has not been taken into account in the previous works; the influence of these variables in the prediction models will be seen in Sec. 5.1. The use of multiple variables is very important since for each working point (Sec. 4.2) the selection of the more representative variables is different. Another difference is the scope of the study: all of the previous works have generally tested their proposal in a single working point.

3 Our Proposal

Over the last few years, our work has been concerned with biometric recognition with successful results [16,9]. These works approach technological aspects of the system, being focused on the feature extraction and recognition stages. However, the maturation of the recognition systems, now ready for practical applications, has encouraged us to go into the final decision stage in greater depth, proposing a system based on MLR for client threshold estimation, whose main characteristics are shown here.

3.1 The Multiple Linear Regression Model

Given a dependent variable, Y, and a set of k explanatory variables $(W_1, \ldots, W_k)$, a MLR represents the relationship between both, as shown in equation: $Y_i = \beta_0 + \sum_{j=1}^{k} \beta_j W_{ij} + \epsilon_i$, where, β_0 is the constant term, β_j are the coefficients relating the k explanatory variables to the variable of interest Y, and ϵ is the random error.

This analysis does not allow us to make causal inferences, but it does allow us to investigate how a set of explanatory variables is associated with a dependent variable of interest. The MLR model is based on several assumptions such as residual normality, homoscedasticity, etc. [13,4], that have been born in mind. Once the model has been estimated by least squares, the regression residuals are calculated as $\hat{\epsilon}_i = \hat{Y}_i - Y_i$, where Y_i is the i observed value of Y, and $\hat{Y}_i$ is the i predictand value. The mean magnitude of relative error $(MMRE)$ and $Pred(R)$ are the most commonly used precision measures. Nevertheless, [11,6] show that $MMRE$ and $Pred(R)$ really estimate the characteristics of $Z_i = \hat{Y}_i / Y_i$, so the use of Z is proposed. Following this, we have used estimations of this variable, such as its Confidence Interval for the mean, to evaluate the prediction model accuracy. Furthermore, we have tested the hypothesis about its “optimal” mean value, $H_0 : \mu_Z = 1$.

Here, this model is proposed to predict the client threshold, i.e: $\hat{\theta}_C = \beta_0 + \sum_{j=1}^{k} \beta_j W_{Cj}$, and $Z_C = \hat{\theta}_C / \theta_C$.

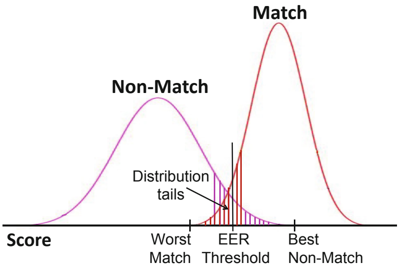


Fig. 1. Match and Non-Match example distributions. The main features referenced in the work have been included.

3.2 Independent Variables

Simple but representative independent variables have been selected. These variables have been estimated using Match and Non-Match sample distributions, that have been achieved by means of the scores from the client and cohort galleries, respectively (Sec. 4.1). A better visualization of this part can be seen in Fig. 1. The selected variables are the following. W_1 = A priori Equal Error Rate (EER) (see Sec. 4.2) threshold achieved using Match and Non-Match sample distributions. W_2 = Client Match distribution mean ($\hat{\mu}_C^M$). W_3 = Client Match standard deviation ($\hat{\sigma}_C^M$). W_4 = Client Match relative standard deviation ($\hat{\sigma}'_{C^M} = \frac{\hat{\sigma}_C^M}{\hat{\mu}_C^M}$). W_5 = Client Non-Match distribution mean ($\hat{\mu}_C^N$). W_6 = Client Non-Match standard deviation ($\hat{\sigma}_C^N$). W_7 = Client Non-Match relative standard deviation ($\hat{\sigma}'_{C^N} = \frac{\hat{\sigma}_C^N}{\hat{\mu}_C^N}$). W_8 = Best Non-Match score. W_9 = Worst Match score. W_{10} = Cohort Gallery or Client Gallery score just less than the a priori EER threshold. W_{11} = Score from Cohort Gallery or Client Gallery just greater than the a priori EER threshold. W_{12} = Client Match scores 25th percentile. W_{13} = Client Match scores 50th percentile. W_{14} = Client Match scores 75th percentile. W_{15} = Client Non-Match scores 25th percentile. W_{16} = Client Non-Match scores 50th percentile. W_{17} = Client Non-Match scores 75th percentile. W_{18} = A priori P_{miss} fixed value (Sec. 4.2) threshold. W_{19} = P_{fa} value for a fixed P_{miss} value.

Variables W_2 to W_7 are based on the assumption that the score distributions are gaussian. W_8 to W_{11} include information about the tail of the distributions, in the same way as the 25th and 75th percentiles. W_{12} to W_{17} are not based on the assumption that the score distributions are gaussian. W_{18} and W_{19} are only used when the threshold for a fixed P_{miss} value is estimated (Sec. 4.2).

3.3 Independent Variables Selection

A correct selection of the variables [1] is very important to achieve an optimal setting of the regression model. To achieve this, a working point dependent selection of the independent variables has been performed.

A stepwise regression has been accomplished by means of the forward selection implemented in the Statgraphics software. The set is initialized with no variables, trying out the variables one by one and including them in the selected set if they are "statistically significant" (R^2 is significantly improved).

4 Experimental Setup

In [9] can be seen the main characteristics of the signature recognition system used here. This was used to participate in the BSEC'2009 signature recognition evaluation, it being the second best system. The classifier is based on the Dynamic Time Warping (DTW) algorithm that has shown a very good performance in the task; DTW performs a distance calculation between the test signature and each of the training ones, being the final score the minimum of these distances. The most popular database in signature recognition has been used: MCYT, with 333 users and 25 authentic and 25 skilled forgery signatures each.

4.1 Experimental Sets

The corpus was split into the following different random subsets:

- **Cohort Set (ChS)**. Consisting of 50 signatories. An authentic signature is randomly selected from each one to get the Cohort Gallery, from which Non-Match sample distribution is estimated.
- **Prediction Model Training Set (PMTrS)**, used to estimate the parameters of the MLR model. To test the model accuracy dependence with regard of this set, two different sizes were taken: i) PMTrS-50 with 50 signatories and ii) PMTrS-100 with 100 signatories.
- **Prediction Model Test Set (PMTeS)**, used to test the prediction model accuracy. Consisting of the signatories not used in the previous sets, that is: i) 233 signatories for PMTrS-50 and ii) 183 signatories for PMTrS-100.

To get the **Client Gallery**, i.e., samples used to get the Match sample distribution, to be completely realistic, only the training samples of each signatory were used. Two techniques can be applied: resubstitution and rotation. In the resubstitution method, each signature is a model of the user, and the distances with regard to the remaining training signatures are calculated; as 5 signatures are used for training, we have 10 scores to estimate the Match distribution. Rotation is implemented by the leave-one-out technique, so the number of scores to estimate the Match distribution are 5. In [17] rotation showed a better performance and, besides, it has fewer distance calculations, so only this technique is used here.

4.2 Working Points

The system working point is completely dependent on the application. Here, we try to test if our proposal can be used to predict different thresholds. Thresholds related with the most used working points to measure system performance in standard evaluations have been estimated: i) The Equal Error Rate (the error of the system when the False Match Rate, P_{fa}, equals the False Non Match Rate, FNMR or P_{miss}), threshold, ii) $P_{miss} = 0$ threshold and iii) $P_{miss} = 0.1$ threshold, i.e., the last two are thresholds for fixed values of FNMR.

5 Experiments

The results achieved with PMTrS-50 and PMTrS-100 were very similar, then, for a more clarity in the exposition the results achieved with PMTrS-50 will be only shown here.

5.1 Model Fitting

The performance of our proposal is evaluated by means of the calculation of the predictive accuracy of the obtained models. These regression equations are

Table 1. Models for each system working point with PTRS-50

Threshold to predict	RND/ SKI	R-squared	Model
EER	RND	91.51	$\hat{\theta}_C = 4.04 - 0,81 * W_3 + 0.47 * W_8 + 0.72 * W_{12}$
EER	SKI	88.75	$\hat{\theta}_C = 10.63 - 188.06 * W_4 + 0.45 * W_8 + 0.68 * W_{14}$
$P_{miss} = 0.0$	RND	95.54	$\hat{\theta}_C = -13.50 + 0.53 * W_{12} + 0.51 * W_{15}$
$P_{miss} = 0.0$	SKI	88.87	$\hat{\theta}_C = -24.05 + 0.54 * W_{14} + 0.53 * W_{15}$
$P_{miss} = 0.1$	RND	99.47	$\hat{\theta}_C = 6.24 + 0.09 * W_8 + 0.24 * W_{15} + 0.36 * W_{16} + 0.25 * W_{19}$
$P_{miss} = 0.1$	SKI	82.09	$\hat{\theta}_C = -17.94 + 1.01 * W_{19}$

Table 2. Testing the models: Results for $Z = \hat{\theta}/\theta$ using PMTeS-233

Threshold to predict	RND/ SKI	R^2	Confidence Interval (95%)	$H_0 : \mu_Z = 1$ $H_1 : \mu_Z \neq 1$ (P-value)	Max Error (%)
EER	RND	96.00	[1.0004; 1.0293]	0,0448	2.9
EER	SKI	93.41	[0.9845; 1.0262]	**0.6167**	-
$P_{miss} = 0.0$	RND	97.70	[1.0368; 1.0751]	2.62E-08	7.5
$P_{miss} = 0.0$	SKI	92.98	[1.0306; 1.0867]	5.34213E-05	8.7
$P_{miss} = 0.1$	RND	99.75	[1.0014; 1.0088]	0.0073	0.9
$P_{miss} = 0.1$	SKI	86.45	[1.0103; 1.0741]	0.0097	7.4

fitted with the Prediction Model Training Set (PMTrS-50) and its predictive power tested with the Prediction Model Test Set (PMTeS-233) (Sec. 4.1).

The obtained models (Sec. 3) and their R-squared are shown in Table 1, for statistical considerations we can accept all of them. In this table, the influence of the variables introduced to model the tail of the distributions can be seen (Sec. 3.2): some of them have been selected in all of the models. Table 2 shows the accuracy of each model by means of some statistics about the threshold prediction ($\hat{\theta}$) with respect to the observed value (θ) using the test set. The p-value of that Table is calculated to test the null Hypothesis that studies the mean value of $Z = \hat{\theta}/\theta$, μ_Z. At best, Z will have 1 as mean value. When the null Hypothesis ($H_0 : \mu_Z = 1$) is rejected, the column called *Max Error (%)* shows the maximum difference between 1 and any value of the Confidence Interval for the mean (maximum prediction error). The null Hypothesis (bold face emphasized in the table) is not rejected in one of the experiments, and it is rejected in five experiments, but with an error of 8.7% at most. The R^2 is high in all cases. So, we can accept the predictions achieved from all of the models.

5.2 Performance Comparison

In order to evaluate the goodness of our proposal, we have compared these results with the state of the art: equations 3, 4, 5 and 6. These equations are fitted with the Prediction Model Training Set (PMTrS-50), minimizing the mean quadratic error ($\sum_C (\theta_C - \hat{\theta}_C)^2/50$), as in our proposal. Their predictive power is tested with the Prediction Model Test Set (PMTeS-233), and measured by means of

Table 3. Performance Comparison of our proposal and the state of the art. D(%) is the relative Difference: $((MMRE_{Eq.(n)} - MMRE_{Our})/MMRE_{Eq.(n)}) * 100$; the results in the table have been calculated without rounded. $\overline{\mathbf{D}}_{\mathbf{c}}(\%)$ is the average of the differences per column and $\overline{\mathbf{D}}_{\mathbf{r}}(\%)$ is the average per row (per Eq.(n)).

	EER				$\mathbf{P_{miss} = 0.0}$				$\mathbf{P_{miss} = 0.1}$				
	RND		**SKI**		**RND**		**SKI**		**RND**		**SKI**		
	MMRE	D(%)	MMRE	D(%)	MMRE	D(%)	MMRE	D(%)	MMRE	D(%)	MMRE	D(%)	$\overline{\mathbf{D}}_{\mathbf{r}}(\%)$.
Our	0.088		0.131		0.098		0.164		0.022		0.203		
Eq.(3)	0.120	27.2	0.170	22.6	0.102	3.8	0.177	7.7	0.028	23.3	0.230	11.8	16.1
Eq.(4)	0.152	42.6	0.207	36.5	0.157	37.4	0.221	26.2	0.078	72.1	0.309	34.2	41.5
Eq.(5)	0.105	16.4	0.151	13.2	0.098	0.2	0.160	-2.4	0.032	32.8	0.226	10.1	11.7
Eq.(6)	0.268	67.4	0.266	50.6	0.262	62.5	0.264	38.1	0.399	94.6	0.339	40.1	58.9
$\overline{\mathbf{D}}_{\mathbf{c}}(\%)$		38.4		30.7		26.0		17.4		55.7		24.1	

$MMRE$ $(\sum_C(|\theta_C - \hat{\theta}_C|/\theta_C)/233)$, as in our proposal again, for an objective comparison. Table 3 shows the results.

Focusing on the $\overline{D}_c(\%)$ row and $\overline{D}_r(\%)$ column (Table 3), it can be seen that our proposal outperforms the rest ones.

Our proposal has achieved the best prediction accuracy for random forgery (RND) in all cases: *EER*, $P_{miss} = 0.0$ and $P_{miss} = 0.1$. With regard to the skilled forgery (SKI), our proposal have obtained the best values, except for $P_{miss} = 0.0$ with Eq. 5, but with very small difference (2.4%).

The best average performance of the state of the art proposals is achieved with the Eq. 5, here. Comparing with this, our proposal performs closely for $P_{miss} = 0.0$, but for *EER* and $P_{miss} = 0.1$, our proposal outperforms Eq. 5 results clearly (18.1% on average).

6 Conclusions and Future Works

In this work a new methodology, based on MLR for a priori client threshold estimation in biometric person recognition systems, has been shown. This proposal has been used to predict the threshold in biometric signature, and for several working points. The prediction models achieved using MLR have been successfully validated looking at the statistical significance of the regression equation and their precision. A comparison with the state of the art has been also performed, showing the advantages, in system performance, of our proposal.

These results are very promising, encouraging us to follow studying in depth the methodology proposed, using new independent variables, with new biometrics. Another interesting subject related with biometric systems is to predict the performance of the user ("biometric menagerie" classification); it would be very interesting to test our proposal in this task. Due to the importance of the estimation of both threshold and performance in real systems, we think that our proposal has very useful future applications.

References

1. Alonso-Gonzalez, C.J., Moro-Sancho, Q.I., Simon-Hurtado, A., Varela-Arrabal, R.: Microarray Gene Expression Classification with Few Genes: Criteria to Combine Attribute Selection and Classification Methods. Expert Systems with Applications 39, 7270–7280 (2012)
2. Chen, K.: Towards Better Making a Decision in Speaker Verification. Pattern Recognition 36, 329–346 (2003)
3. Doddington, G.R.: Speaker Recognition Evaluation Methodology an Overview and Perspective. In: Proceedings of the RLA2C Workshop on Speaker Recognition and its Commercial and Forensic Applications, Avignon, France, pp. 60–66 (1998)
4. Draper, N.R., Smith, H.: Applied Regression Analysis. Wiley, Massachusetts (1981)
5. Faundez-Zanuy, M.: On-line Signature Recognition Based on VQ-DTW. Pattern Recognition 40, 981–992 (2006)
6. Foss, T., Stensrud, E., Kitchenham, B.I.M.: A Simulation Study of the Model Evaluation Criterion MMRE. IEEE Transactions on Software Ingineering 29, 985–995 (2003)
7. Furui, S.: Cepstral Analysis Technique for Automatic Speaker Verification. IEEE Transactions on ASSP 29, 254–272 (1981)
8. Hernando, D., Saeta, J., Hernando, J.: Threshold Estimation with Continuously Trained Models in Speaker Verification. In: The Speaker and Language Recognition Workshop, pp. 1–4, Odyssey (2006)
9. Houmani, N., Mayoue, A., Garcia-Salicetti, S., Dorizzi, B., Khalil, M., Moustafa, M., Abbas, H., Muramatsu, D., Yanikoglu, B., Kholmatov, A., Martinez-Diaz, M., Fierrez, J., Ortega-Garcia, J., Alcobe, J.R., Fabregas, J., Faundez-Zanuy, M., Pascual-Gaspar, J., Cardeñso-Payo, V., Vivaracho-Pascual, C.: Biosecure Signature Evaluation Campaign (BSEC 2009): Evaluating Online Signature Algorithms Depending on the Quality of Signatures. Pattern Recognition 45, 993–1003 (2012)
10. International Biometric Group: Biometrics Market and Industry Report (2009-2014), http://www.ibgweb.com/products/reports/bmir-2009-2014 (last visited: August 2012)
11. Kitchenham, B., Pickard, L., MacDonell, S.G., Shepperd, M.: What Accuracy Statistics really Measure (software estimation). IEE Proceedings-Software 148, 81–85 (2001)
12. Lindberg, J., Koolwaaij, J., Hutter, H., Genoud, D., Pierrot, J., Blomberg, M., Bimbot, F.: Techniques for a Priori Decision Threshold Estimation in Speaker Verification. In: Proceedings of the RLA2C Workshop on Speaker Recognition and Its Commercial and Forensic Applications, Avignon, France, pp. 89–92 (1998)
13. Ostrom, C.W.: Time Series Analysis, Regression Techniques, 2nd edn., Quantitative Applications in the Social Sciences, vol. 07-009. SAGE (1990)
14. Saeta, J., Hernando, J.: Assessment of On-line Model Quality and Threshold Estimation in Speaker Verification. IEICE - Transactions on Information and Systems E990-D(4) (2007)
15. Shi, Z., Kiefer, F., Schneider, J., Govindaraju, V.: Modeling Biometric Systems Using the General Pareto Distribution (GPD). In: Proc. of the SPIE, vol. 6944, pp. 69440O-1–69440O-11 (2008)
16. Vivaracho, C.: ISCSLP SR Evaluation, UVA–CS_es System Description. A System Based on ANNs. In: Huo, Q., Ma, B., Chng, E.-S., Li, H. (eds.) ISCSLP 2006. LNCS (LNAI), vol. 4274, pp. 529–538. Springer, Heidelberg (2006)
17. Vivaracho-Pascual, C., Faundez-Zanuy, M., Pascual, J.M.: An Efficient Low Cost Approach for On-line Signature Recognition Based on Length Normalization and Fractional Distances. Pattern Recognition 42, 183–193 (2009)

Deterministic Annealing Multi-Sphere Support Vector Data Description

Trung Le, Dat Tran*, Wanli Ma, and Dharmendra Sharma

University of Canberra, ACT 2601 Australia
{trung.le,dat.tran,wanli.ma,dharmendra.sharma}@canberra.edu.au

Abstract. Current well-known data description method such as Support Vector Data Description is conducted with assumption that data samples of a class in feature space are drawn from a single distribution. Based on this assumption, a single hypersphere is constructed to provide a good data description for the data. However, real-world data samples may be drawn from some distinctive distributions and hence it does not guarantee that a single hypersphere can offer the best data description. In this paper, we introduce a Deterministic Annealing Multi-sphere Support Vector Data Description (DAMS-SVDD) approach to address this issue. We propose to use a set of hyperspheres to provide a better data description for a given data set. Calculations for determining optimal hyperspheres and experimental results for applying this proposed approach to classification problems are presented.

Keywords: Kernel Methods, Deterministic Annealing , Support Vector Data Description, Multi-Sphere Support Vector Data Description.

1 Introduction

Support Vector Data Description (SVDD) [5] is one of the most well-known method for one-class classification problems. SVDD assumes that all samples of the training set are drawn from a single uniform distribution [5]. However, this hypothesis is not always true since real-world data samples may be drawn from distinctive distributions [6]. Therefore, a single hypersphere cannot be a good data description. For example, in Figure 1, data samples are scattered over some distinctive distributions and one single hypersphere would improperly record the inside outliers. In [6], a multi-sphere approach to SVDD was proposed for multi-distribution data. The domain for each distribution was detected and for each domain an optimal sphere was constructed to describe the corresponding distribution. However, the learning process was heuristic and did not follow up learning with minimal volume principle [4]. In [2], a method was proposed to link the input space to the feature space. Dense regions (clusters) in the input space were identified and became a single sphere in the feature space. Again, this method was heuristic and did not abide by learning with minimum

[*] Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 183–190, 2012.

volume principle. To motivate learning with minimum volume principle, we have proposed a hard multi-sphere support vector data description (HMS-SVDD) [3]. A set of hyperspheres was introduced to enclose all the data samples. A data sample will belong to only one hypersphere. The volume of enclosing shape is minimised to favor generalisation capacity of classifier. However, this method cannot avoid local minima.

Inspired from physical experiments, a new stochastic optimisation strategy, Simulated Annealing or Deterministic Annealing (DA) was proposed in [1]. Through analogy to an experimental annealing where a stability of metal is improved by heating and cooling, solutions for optimisation problem are heated and cooled in simulation to find one with low costs. DA offers two important features: ability to avoid the local minima, and the capability to find out the minima of right objective function even its gradients almost vanish everywhere. To benefit from DA's advantageous features, we propose in this paper DA approach of Multi-Sphere Support Vector Data Description. A set of hyperspheres is proposed to describe the normal data set assuming that normal data samples have distinctive data distributions. DA approach allows our solution to avoid the local minima when the temperature variable is led to approach 0.

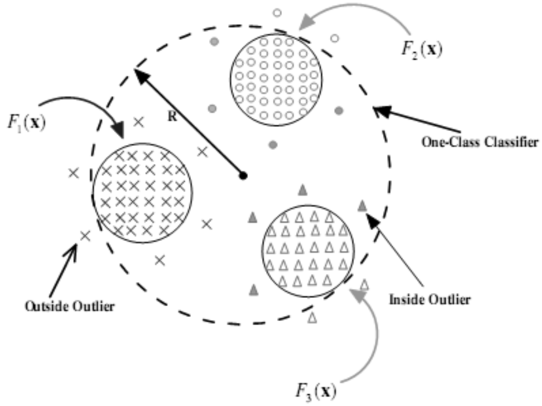


Fig. 1. Inside outliers would be improperly included if only one hypersphere is constructed [6]

2 Deterministic Annealing Multi-Sphere Support Vector Data Description (DAMS-SVDD)

2.1 Problem Formulation

Consider a set of m hyperspheres $S_j(c_j, R_j)$ with center c_j and radius R_j, $j = 1, \ldots, m$. This hypershere set is a good data description of the normal data set $X = \{x_1, x_2, \ldots, x_n\}$ if each of the hyperspheres describes a distribution in this data set and the sum of all radii $\sum_{j=1}^{m} R_j^2$ should be minimised. Let matrix $U = [u_{ij}]_{n\times m}$, $u_{ij} \in \{0,1\}$, $i = 1,\ldots,n$, $j = 1,\ldots,m$ where u_{ij} is the membership representing degree of belonging of sample x_i to hypersphere S_j and $\sum\limits_{j=1}^{m} u_{ij} = 1$. The optimisation problem of DAMS-SVDD can be formulated as follows

$$\min_{R,c,u,\xi} \Big(\sum_{j=1}^{m} R_j^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i \Big) \tag{1}$$

$$s.t.: \sum_{j=1}^{m} u_{ij}||\phi(x_i) - c_j||^2 \leq \sum_{j=1}^{m} u_{ij} R_j^2 + \xi_i;\ \xi_i \geq 0, \quad i = 1,\ldots,n \tag{2}$$

where $R = [R_j]_{j=1,\ldots,m}$ is vector of radii, ν is a constant, ξ_i are slack variables, $\phi(.)$ is a transformation from input space to feature space, and $c = [c_j]_{j=1,\ldots,m}$ is vector of centres.

Minimising the function in (1) over variables R, c and ξ subjecting to (2) will determine radii and centres of hyperspheres and slack variables and the partition matrix U as well. These quantities contribute to form the decision boundary.

For classifying a sample x, the following decision function is used

$$f(x) = sign\Big(\max_{1\leq j\leq m} \Big\{ R_j^2 - ||\phi(x) - c_j||^2 \Big\} \Big) \tag{3}$$

We present in the next sections the way to apply DA to resolve the above OP (1). The coming solution takes advantage from the capability to converge to global optimal solution of DA.

2.2 DA Optimization Problem Extension

It is shown that the number of partition matrices U is m^n. This huge number circumvents a brute-force strategy to figure out the optimal solution. Fortunately, by chance DA can be applied to the above problem. We extend the above OP by replacing u_{ij} by p_{ij} that varies in $[0;1]$ and can be interpreted as probability where x_i belongs to hypersphere S_j. The new OP is as follows

$$\begin{aligned} &\min_{R,c,p,\xi} \left(\sum_{j=1}^{m} R_j^2 + \tfrac{1}{\nu n} \sum_{i=1}^{n} \xi_i + T \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} \ln p_{ij} \right) \\ &s.t.: \sum_{j=1}^{m} p_{ij} \left\| \phi(x_i) - c_j \right\|^2 \leq \sum_{j=1}^{m} p_{ij} R_j^2 + \xi_i;\ \sum_{j=1}^{m} p_{ij} = 1; \xi_i \geq 0,\ i = 1,\ldots,n \end{aligned} \tag{4}$$

where $T > 0$ is temperature variable.

2.3 Solution

Given T, we alternately keep fixed R, c and p respectively to find out the new hypersphere set and probability partition matrix respectively. *Kullback-Leibler divergence* (KL-divergence) is used as stopping criterion for algorithm.

Fixed p. By eliminating the constants, we achieve the following OP

$$\begin{aligned}&\min_{R,c}\left(\sum_{j=1}^{m}R_j^2+\frac{1}{\nu n}\sum_{i=1}^{n}\xi_i\right)\\&s.t.:\ \sum_{j=1}^{m}p_{ij}\|\phi(x_i)-c_j\|^2\le\sum_{j=1}^{m}p_{ij}R_j^2+\xi_i;\ \xi_i\ge 0,\quad i=1,\dots,n\end{aligned}\tag{5}$$

To deal with the above OP, we make use of *Karush-Kuhn-Tucker (KKT) theorem.* The Lagrange function is of the following form

$$\begin{aligned}&L(R,c,\xi,\alpha,\beta)=\sum_{j=1}^{m}R_j^2+\frac{1}{\nu n}\sum_{i=1}^{n}\xi_i\\&+\sum_{i=1}^{n}\alpha_i\left(\sum_{j=1}^{m}p_{ij}\|\phi(x_i)-c_j\|^2-\sum_{j=1}^{m}p_{ij}R_j^2-\xi_i\right)-\sum_{i=1}^{n}\beta_i\xi_i\end{aligned}\tag{6}$$

Setting derivatives to 0, we obtain

$$\begin{aligned}&\tfrac{\delta L}{\delta R_j}=0\Rightarrow 1=\sum_{i=1}^{n}p_{ij}\alpha_i,\ j=1,\dots,m\\&\tfrac{\delta L}{\delta c_j}=0\Rightarrow c_j=\sum_{i=1}^{n}p_{ij}\alpha_i\phi(x_i),\ j=1,\dots,m\\&\tfrac{\delta L}{\delta \xi_i}=0\Rightarrow \alpha_i+\beta_i=\tfrac{1}{\nu n},\ i=1,\dots,n\end{aligned}\tag{7}$$

By substituting (7 to Lagrange function, we obtain the dual form

$$\begin{aligned}&L(R,c,\xi,\alpha,\beta)=\sum_{i=1}^{n}\alpha_iK(x_i,x_i)+\sum_{i=1}^{n}\sum_{j=1}^{m}p_{ij}\alpha_i\left(\|c_j\|^2-2\phi(x_i)c_j\right)\\&=\sum_{i=1}^{n}\alpha_iK(x_i,x_i)+\sum_{i=1}^{n}\sum_{j=1}^{m}p_{ij}\alpha_i\|c_j\|^2-2\sum_{i=1}^{n}\sum_{j=1}^{m}p_{ij}\alpha_i\phi(x_i)c_j\\&=\sum_{i=1}^{n}\alpha_iK(x_i,x_i)+\sum_{j=1}^{m}\sum_{i=1}^{n}p_{ij}\alpha_i\|c_j\|^2-2\sum_{j=1}^{m}\sum_{i=1}^{n}p_{ij}\alpha_i\phi(x_i)c_j\\&=\sum_{i=1}^{n}\alpha_iK(x_i,x_i)+\sum_{j=1}^{m}\|c_j\|^2-2\sum_{j=1}^{m}\|c_j\|^2=-\sum_{i,i'}p_ip_{i'}K(x_i,x_{i'})\alpha_i\alpha_{i'}+\sum_{i=1}^{n}\alpha_iK(x_i,x_i)\end{aligned}\tag{8}$$

where $p_i=[p_{i1},p_{i2},...,p_{im}]$ and $p_ip_{i'}=\sum_{j=1}^{m}p_{ij}p_{i'j}$.

Therefore, we come up with the following quadratic OP

$$\begin{aligned}&\min_{\alpha}\left(\sum_{i,i'}p_ip_{i'}K(x_i,x_{i'})\alpha_i\alpha_{i'}-\sum_{i=1}^{n}\alpha_iK\left(x_i,x_i\right)\right)\\&s.t.:\ \sum_{i=1}^{n}p_{ij}\alpha_i=1;\ 0\le\alpha_i\le\tfrac{1}{\nu n},\ i=1,\dots,n\end{aligned}\tag{9}$$

Fixed *R*, *c*. By removing the constants, we gain the following OP

$$\begin{aligned}&\min_{p}\left(\frac{1}{\nu n}\sum_{i=1}^{n}\max\left\{0,\sum_{j=1}^{m}p_{ij}d_{ij}\right\}+T\sum_{i=1}^{n}\sum_{j=1}^{m}p_{ij}\ln p_{ij}\right)\\&s.t:\sum_{j=1}^{m}p_{ij}=1,\ i=1,\dots,n\ and\ p_{ij}\ge 0,\ i=1,\dots,n,\ j=1,\dots,m\end{aligned}\tag{10}$$

where $d_{ij} = \|\phi(x_i) - c_j\|^2 - R_j^2, \quad i = 1, \ldots, n,\ j = 1, \ldots, m$, and $P = [p_{ij}]_{n \times m}$ is probability partition matrix.

It is obvious that we can separate the above OP to n individual OPs as follows

$$\begin{aligned} &\min_{p_i} \left(\tfrac{1}{\nu n} \max \left\{ 0, \sum_{j=1}^{m} p_{ij} d_{ij} \right\} + T \sum_{j=1}^{m} p_{ij} \ln p_{ij} \right) \\ &s.t.: \sum_{j=1}^{m} p_{ij} = 1 \end{aligned} \tag{11}$$

To derive the optimisation problem (11), let us introduce two sets

$$\begin{aligned} A^{+} &= \left\{ p_i = [p_{i1}, ..., p_{im}] : \sum_{j=1}^{m} p_{ij} d_{ij} \geq 0 \; and \; p_{ij} \geq 0 \; for \; all \; j \right\} \\ A^{-} &= \left\{ p_i = [p_{i1}, ..., p_{im}] : \sum_{j=1}^{m} p_{ij} d_{ij} < 0 \; and \; p_{ij} \geq 0 \; for \; all \; j \right\} \end{aligned} \tag{12}$$

We examine two possible cases

i) $\mathbf{p_i} \in \mathbf{A}^{+}$

The optimisation problem (11) becomes

$$\begin{aligned} &\min_{p_i} \left(C \sum_{j=1}^{m} p_{ij} d_{ij} + T \sum_{j=1}^{m} p_{ij} \ln p_{ij} \right) \\ &s.t.: \sum_{j=1}^{m} p_{ij} = 1 \; and \; \sum_{j=1}^{m} p_{ij} d_{ij} \geq 0 \end{aligned} \tag{13}$$

Again, we apply KKT theorem to cope with this optimisation. The Lagrange function is of the following form

$$L(p_i, \alpha, \lambda) = C \sum_{j=1}^{m} p_{ij} d_{ij} + T \sum_{j=1}^{m} p_{ij} \ln p_{ij} + \lambda \left(\sum_{j=1}^{m} p_{ij} - 1 \right) - \alpha \sum_{j=1}^{m} p_{ij} d_{ij} \tag{14}$$

Setting derivatives to 0, we gain

$$\begin{aligned} &\tfrac{\partial L}{\partial p_{ij}} = 0 \Rightarrow C d_{ij} + T(1 + \ln p_{ij}) + \lambda - \alpha d_{ij} = 0 \Rightarrow p_{ij} = e^{\frac{\alpha - C}{T} d_{ij} - \frac{\lambda}{T} - 1} \\ &\sum_{j=1}^{m} p_{ij} = 1; \; \sum_{j=1}^{m} p_{ij} d_{ij} \geq 0; \; \alpha \geq 0; \; \alpha. \left(\sum_{j=1}^{m} p_{ij} d_{ij} \right) = 0 \end{aligned} \tag{15}$$

From (15), we have $p_{ij}(\alpha) = \frac{e^{\frac{\alpha - C}{T} d_{ij}}}{\sum_{j'=1}^{m} e^{\frac{\alpha - C}{T} d_{ij'}}}$. To examine the equation: $\alpha. \left(\sum_{j=1}^{m} p_{ij}(\alpha) d_{ij} \right) = 0$, we define and investigate the function $f(x) = \sum_{j=1}^{m} e^{\frac{x - C}{T} d_{ij}} d_{ij}$ where $x \geq 0$. We have $f'(x) = \frac{1}{T} \sum_{j=1}^{m} e^{\frac{x - C}{T} d_{ij}} d_{ij}^2 > 0$. It discloses that $f(x)$ is a strictly increasing function. We branch out two cases.

□ *Case 1* ($f(0) = \sum\limits_{j=1}^{m} e^{\frac{-C}{T}d_{ij}} d_{ij} \geq 0$): It means that $f(x) > 0$, if $x > 0$. Hence the equation $\alpha f(\alpha) = 0$ has got unique solution $\alpha = 0$ and the solution for optimisation problem (13) is: $p_{ij} = p_{ij}(0) = \frac{e^{\frac{-C}{T}d_{ij}}}{\sum\limits_{j'=1}^{m} e^{\frac{-C}{T}d_{ij'}}}$.

□ *Case 2* ($f(0) = \sum\limits_{j=1}^{m} e^{\frac{-C}{T}d_{ij}} d_{ij} < 0$): If $d_{ij} < 0$ for all j then $A^+ = \phi$ and i) cannot happen. Otherwise, we have $\lim\limits_{x \to +\infty} f(x) = +\infty$. Therefore, there exists unique $\alpha_1 > 0$ such that $f(\alpha_1) = 0$. In practice, we specify the unique solution of equation $f(\alpha) = 0$ by Newton-Raphson method.

ii) $\mathbf{p_i} \in \mathbf{A}^-$

Similar to the first case, we can fulfill the derivation. This derivation relates to function $g(x) = \sum\limits_{j=1}^{m} e^{\frac{-x}{T}d_{ij}} d_{ij}$ and as follows

□ *Case 1* ($g(0) = \sum\limits_{j=1}^{m} d_{ij} < 0$): $p_{ij} = p_{ij}(0) = \frac{1}{m}$ for all j.

□ *Case 2* ($g(0) = \sum\limits_{j=1}^{m} d_{ij} \geq 0$): $p_{ij} = \frac{e^{\frac{-\alpha_2}{T}d_{ij}}}{\sum\limits_{j'=1}^{m} e^{\frac{-\alpha_2}{T}d_{ij'}}}$ where α_2 is the unique solution of equation $g(\alpha) = 0$ which can be specified by Newton-Raphson method.

2.4 The Overall Algorithm (DAMS-SVDD)

We start with $T = 10$. For each T, we attempt to solve out the OP in (4) by alternately keeping R, c and p fixed. The *KL-divergence* is used as stopping criterion for each iteration. To direct the local minimizer attained for each T to the global minimizer, T is led to approach 0. The detail of this algorithm is displayed in the following.

```
Initialize
    $T = 10, \varepsilon = 0.001$
     for($i = 1$; $i \leq n$; $i++$) Set $p_{ij} = \frac{1}{m}$ for all $j = 1, \ldots, m$
    $q = p$
Execute
     while($T > \varepsilon$){
      while($D_{KL}(p, q) > \varepsilon$){
        Keep fixed $p$
          Calculate $R = [R_1, R_2, ..., R_m]$, $c = [c_1, c_2, ..., c_m]$, and
          $d_{ij} = \|\phi(x_i) - c_j\|^2 - {R_j}^2$ as in subsection fixed $p$.
        Keep fixed $R, c$
          $q = p$
          Calculate probability partition matrix $p$ as in subsection fixed $R, c$
        }
    $T = \frac{T}{1.5}$
}
```

where $D_{KL}(p,q) = \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij} \ln \frac{p_{ij}}{q_{ij}}$

Table 1. The experimental results on 13 data sets

Datasets	*SVDD*	*HMS-SVDD*	*DAMS-SVDD*
Australian	88%	92%	**95%**
Breast cancer	90%	96%	**100%**
Astroparticle	94%	91%	**98%**
Dna	90%	88%	86%
Sonar	79%	76%	**88%**
Inosphere	87%	96%	94%
Bioinformatics	85%	99%	98%
Diabetes	90%	98%	97%
Liver	88%	92%	90%
Splice	86%	90%	**93%**
Vehicle	89%	88%	**96%**
Delp Pump	89%	95%	**98%**
USPS	93%	94%	**96%**

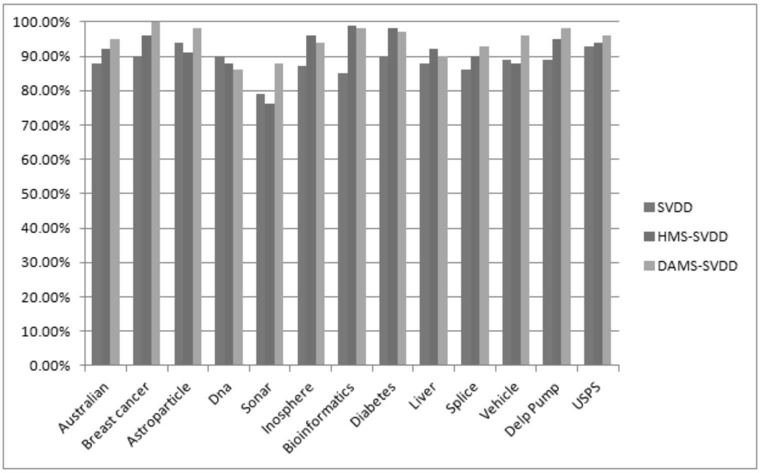


Fig. 2. Experimental results on 13 data sets

3 Experiment

To show the performance of the proposed model, we established experiment on 13 data sets of UCI repository. For each data set, we selected one class and appointed it as normal class. We ran cross validation with five folds. DAMS-SVDD was compared with the most well-known kernel-based one-class classification method SVDD [5] and HMS-SVDD [3].

The popular RBF Kernel $K(x, x') = e^{-\gamma\|x-x'\|^2}$ was applied whereas the parameter γ is varied in grid $\{2^i : i = 2j+1, j = -8, \ldots, 1\}$. The parameter ν was selected in grid $\{0.1i : i = 1, \ldots, 9\}$. The number of hyperspheres m was searched in grid $\{2, 3, 5, 7, 9\}$.

The attained result as shown in Table 1 and Figure 2 illustrates that DAMS-SVDD outperforms the other methods especially for the data sets with multiple distributions.

4 Conclusion

We propose DA solution for multi-sphere approach to SVDD. A set of hyperspheres is learnt rather than a single sphere. The solution can take advantage of DA to avoid local minima and converge to the global optimum when *temperature* T is approached 0. The experiment established on 13 data sets shows that the DAMS-SVDD can provide good data description especially for data sets with multiple distributions.

References

1. Aarts, E., Korst, J.: Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing, 1st edn. Wiley (1988)
2. Chiang, J.H., Hao, P.Y.: A New Kernel-Based Fuzzy Clustering Approach:Support Vector Clustering with Cell Growing. IEEE T. Fuzzy Syst. 11, 518–527 (2003)
3. Le, T., Tran, D., Ma, W., Sharma, D.: A Theoretical Framework for Multi-sphere Support Vector Data Description. In: Proceedings of the 17th International Conference on Neural Information Processing: Models and Applications, pp. 132–142. Springer, Berlin (2010)
4. Scott, C.D., Nowak, R.D.: Learning Minimum Volume Sets. J. Mach. Learn. Res. 7, 665–704 (2006)
5. Tax, D.M.J., Duin, R.P.W.: Support Vector Data Description. J. Mach. Learn. Res. 54, 45–66 (2004)
6. Xiao, Y., Liu, B., Cao, L., Wu, X., Zhang, C., Hao, Z., Yang, F., Cao, J.: Multi-sphere Support Vector Data Description for Outliers Detection on Multi-distribution Data. In: ICDM Workshops, pp. 82–87 (2009)

Maximal Margin Approach to Kernel Generalised Learning Vector Quantisation for Brain-Computer Interface

Trung Le, Dat Tran*, Tuan Hoang, and Dharmendra Sharma

Faculty of Information Sciences and Engineering
University of Canberra, ACT 2601, Australia
{trung.le,dat.tran,tuan.hoang,dharmendra.sharma}@canberra.edu.au

Abstract. Kernel Generalised Learning Vector Quantisation (KGLVQ) was proposed to extend Generalised Learning Vector Quantisation into the kernel feature space to deal with complex class boundaries and thus yield promising performance for complex classification tasks in pattern recognition. However KGLVQ does not follow the maximal margin principle which is crucial for kernel-based learning methods. In this paper we propose a maximal margin approach to Kernel Generalised Learning Vector Quantisation algorithm which inherits the merits of KGLVQ and follows the maximal margin principle to favour the generalisation capability. Experiments performed on the well-known data set III of BCI competition II show promising classification results for the proposed method.

Keywords: Learning Vector Quantisation, Generalised Learning Vector Quantisation, Kernel Method, Maximising Margin.

1 Introduction

Self-organizing methods such as the Self-Organizing Map (SOM) or Learning Vector Quantisation (LVQ) introduced by Kohonen [7] provide a successful and intuitive method of processing data for easy access [5]. Learning Vector Quantisation (LVQ) aims at generating the prototypes or reference vectors which delegate for the data of classes [6]. Although LVQ is a fast and simple learning algorithm, sometimes its prototypes diverge and as a result degrade recognition ability [12]. To address this problem, Generalised Learning Vector Quantisation (GLVQ) [12] was proposed. It is a generalisation of the original model proposed by Kohonen and the prototypes are updated based on the steepest descent method to minimise a cost function. GLVQ has been widely applied and shown good performance in many applications [8], [11], [12]. However, its performance may deteriorate for complex data sets since pattern classes with nonlinear class boundaries usually need a large number prototypes; but when we require a large number of prototypes in the input space, it can be difficult to determine the reasonable number and their positions while achieving a good

* Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 191–198, 2012.

generalisation performance [10]. To overcome this drawback, in [10] Kernel Generalised Learning Vector Quantisation (KGLVQ) was proposed for learning the prototypes of data in feature space. Like LVQ and GLVQ, KGLVQ can be used for two class and multi-class classification problems. In case of two-class classification problem, the entire feature space would be divided into subspaces induced by two core prototypes and in each subspace a mid-perpendicular hyperplane of two these core prototypes was employed to classify the data. Nevertheless, these induced hyperplanes of KGLVQ do not guarantee maximising margins which is crucial for kernel methods [13].

In this paper, we propose a maximal margin approach to KGLVQ which takes advantage of maximising margins for increasing the generalisation capability as seen in Support Vector Machine [1], [2]. Our proposed approach is different from the approach in [3] which aims at maximising the hypothesis margin rather than the real margin. In our approach, a finite number of prototypes m and n are used to represent positive and negative classes, respectively in binary data sets. The entire feature space is divided into $m \times n$ subspaces induced by pairs of prototypes and in each subspace a mid-perpendicular hyperplane of two correspondent prototypes is employed to classify the data. The cost function in our approach takes into account maximising the margins of hyperplanes to boost the generalisation capability. Experiment on the well-known data set III of Brain-Computer Interface (BCI) competition II was established. The results show the prospective for applying the proposed method to BCI data because of its performance and simplicity.

2 Maximal Margin Kernel Generalised LVQ

2.1 Introduction

Consider a binary training set $X = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$ where $x_1, x_2, \ldots, x_l \in \mathrm{I\!R}^d$ are points and $y_1, y_2, \ldots, y_l \in \{-1, 1\}$ are labels. This training set is mapped into a high dimensional space namely feature space through a function $\phi(.)$. Based on the idea of Vector Quantisation (VQ), m prototypes $A_1, A_2, \ldots, A_m$ of the positive class and n prototypes $B_1, B_2, \ldots, B_n$ of the negative class will be discovered in the feature space. The decision rule is based on the minimum distance to the prototypes in each class. More precisely, given a new vector x the decision function is as follows

$$f(x) = sign\left(\|\phi(x) - b_{j_0}\|^2 - \|\phi(x) - a_{i_0}\|^2\right) \tag{1}$$

where $i_0 = \underset{1 \le i \le m}{\arg\min} \left\{\|\phi(x) - a_i\|^2\right\}$, $j_0 = \underset{1 \le j \le n}{\arg\min} \left\{\|\phi(x) - b_j\|^2\right\}$, and a_i, b_j are coordinates of A_i, B_j, $i = 1, \ldots, m$; $j = 1, \ldots, n$, respectively.

2.2 Optimisation Problem

Given a labeled training vector (x, y), denote a and b as two prototypes of the positive class and negative class which are closest to $\phi(x)$. Let $\mu(x, a, b)$ be the function which satisfies the following criterion: if x is correctly classified, $\mu(x, a, b) < 0$; otherwise $\mu(x, a, b) \geq 0$. Let g be a monotonically increasing function. To reduce the error rate, $\mu(x, a, b)$ should decrease for all training vectors. The criterion is formulated as minimising of the following function:

$$\min_{\{A\},\{B\}} \sum_{i=1}^{l} g\left(\mu\left(x_i, a^{(i)}, b^{(i)}\right)\right) \tag{2}$$

where $\{A\}$ and $\{B\}$ are the sequences $\{A_1, A_2, \ldots, A_m\}$ and $\{B_1, B_2, \ldots, B_n\}$, respectively, and $a^{(i)}$ and $b^{(i)}$ are two class prototypes which are closest to $\phi(x_i)$.

2.3 Solution

We assume that the prototypes are linear expansions of vectors $\phi(x_1), \ldots, \phi(x_l)$. Denote $a_i,\ i = 1, \ldots, m$ and $b_j,\ j = 1, \ldots, n$ as coordinates of the prototypes:

$$a_i = \sum_{k=1}^{l} u_{ik}\phi(x_k),\ i = 1, \ldots, m \qquad b_j = \sum_{k=1}^{l} v_{jk}\phi(x_k),\ j = 1, \ldots, n \tag{3}$$

For convenience, if $c = \sum_{i=1}^{l} u_i\phi(x_i)$, we rewrite c as $c = [u_1, u_2, \ldots, u_l]$. Given a labeled training vector (x, y), first we determine two closest prototypes A and B for two classes with respect to x, second we use gradient descent method to update the coordinates a and b of A and B, respectively as follows:

$$a = a - \alpha\frac{\partial g}{\partial \mu}\frac{\partial \mu}{\partial a} \qquad b = b - \alpha\frac{\partial g}{\partial \mu}\frac{\partial \mu}{\partial b} \tag{4}$$

```
ALGORITHM FOR VECTOR QUANTISATION
  Initialise
    Using C-Means or Fuzzy C-Means clustering to find $m$ protoypes
    for positive class and $n$ protoypes for negative class in the input space.
    Set $t = 0$ and $i = 0$
  Repeat
    $t = t + 1$
    $i = (i + 1)$ mod $l$
    $A_t = A_{i_0}$ where $i_0 = \arg\min_{1\leq k\leq m}\left\{\|\phi(x_i) - a_k\|^2\right\}$
    $B_t = B_{j_0}$ where $j_0 = \arg\min_{1\leq k\leq n}\left\{\|\phi(x_i) - b_k\|^2\right\}$
    Update $a_{i_0} = a_{i_0} - \alpha\frac{\partial g}{\partial \mu}\frac{\partial \mu}{\partial a_{i_0}}$
    Update $b_{j_0} = b_{j_0} - \alpha\frac{\partial g}{\partial \mu}\frac{\partial \mu}{\partial b_{j_0}}$
  Until convergence is reached
```

where the sigmoid function $g = g(\mu, t)$ depends on learning time t. The function $g(\mu, t) = \frac{1}{1+e^{-\mu t}}$ is a good candidate for g. If this sigmoid function is applied then $\frac{\partial g}{\partial \mu} = tg(\mu, t)(1 - g(\mu, t))$.

2.4 Selection of the μ-function

We introduce some candidates for the μ-function. Let (x, y) be a labeled training vector and a & b are two closest prototypes in two classes to that vector.

Candidate 1 for the μ-function [7] (LVQ)

$$\mu(x,a,b) = y\left(\|\phi(x)-a\|^2 - \|\phi(x)-b\|^2\right) = y(d_1-d_2) = \eta(d_1,d_2) \tag{5}$$

Candidate 2 for the μ-function [12] (GLVQ)

$$\mu(x,a,b) = \frac{y\left(\|\phi(x)-a\|^2 - \|\phi(x)-b\|^2\right)}{\|\phi(x)-a\|^2 + \|\phi(x)-b\|^2} = \frac{y(d_1-d_2)}{d_1+d_2} = \eta(d_1,d_2) \tag{6}$$

where d_1 and d_2 in (5) and (6) are distances from $\phi(x)$ to two prototypes a and b, respectively. These functions depend primarily on d_1 and d_2. The formula for adaptation of prototypes in (4) can be rewritten as follows

$$a = a - 2\alpha\tfrac{\partial g}{\partial \eta}\tfrac{\partial \eta}{\partial d_1}\left(a-\phi(x)\right) \qquad b = b - 2\alpha\tfrac{\partial g}{\partial \eta}\tfrac{\partial \eta}{\partial d_2}\left(b-\phi(x)\right) \tag{7}$$

If $\mu(x,a,b) = \eta(d_1,d_2) = y(d_1-d_2)$, the equations in (7) become:

$$a = a - 2\alpha\tfrac{\partial g}{\partial \eta}y\left(a-\phi(x)\right) \qquad b = b + 2\alpha\tfrac{\partial g}{\partial \eta}y\left(b-\phi(x)\right) \tag{8}$$

If $\mu(x,a,b) = \eta(d_1,d_2) = \frac{y(d_1-d_2)}{d_1+d_2}$, the equations in (7) become:

$$a = a - \alpha\tfrac{\partial g}{\partial \eta}\tfrac{4yd_2}{(d_1+d_2)^2}\left(a-\phi(x)\right) \qquad b = b + \alpha\tfrac{\partial g}{\partial \eta}\tfrac{4yd_1}{(d_1+d_2)^2}\left(b-\phi(x)\right) \tag{9}$$

Candidate 3 for the μ-function [3] (HMLVQ)

$$\mu(x,a,b) = \frac{1}{2}y\left(\|\phi(x)-a\| - \|\phi(x)-b\|\right) \tag{10}$$

This μ-function refers to hypothesis margin in [3] and is used in AdaBoost [4]. The hypothesis margin measures how much the hypothesis can travel before it hits an instance as shown in Fig. 1.

The partial derivatives of μ with respect to a and b are:

$$\tfrac{\partial \mu}{\partial a} = -\tfrac{y}{2\|\phi(x)-a\|}\left(\phi(x)-a\right) \qquad \tfrac{\partial \mu}{\partial b} = \tfrac{y}{2\|\phi(x)-b\|}\left(\phi(x)-b\right) \tag{11}$$

Candidate 4 for μ-function (MLVQ and KMLVQ)

This is our proposed maximal margin approach to LVQ. We name it as MLVQ for the model in the input space, and KMLVQ for that in the feature space. The μ-function is of the form

$$\mu(x,a,b) = \frac{y(\|\phi(x)-a\|^2 - \|\phi(x)-b\|^2)}{\|a-b\|} \tag{12}$$

It is noted that the absolute value of the μ-function in Candidate 4 is the sample margin at $\phi(x)$ in Fig. 1, also the distance from $\phi(x)$ to mid-perpendicular hyperplane of prototypes a and b. When x is correctly classified, this value is equal to negative sample margin at x. Minimising $\mu(x, a, b)$ motivates maximising the sample margin at x.

The partial derivatives of μ with respect to a and b are

$$\begin{aligned}\frac{\partial\mu}{\partial a} &= \frac{-2y}{\|a-b\|}\left(\phi(x)-a\right) - \frac{y\left(\|\phi(x)-a\|^2-\|\phi(x)-b\|^2\right)}{\|a-b\|^3}\left(a-b\right)\\ \frac{\partial\mu}{\partial b} &= \frac{2y}{\|a-b\|}\left(\phi(x)-b\right) + \frac{y\left(\|\phi(x)-a\|^2-\|\phi(x)-b\|^2\right)}{\|a-b\|^3}\left(a-b\right)\end{aligned} \tag{13}$$

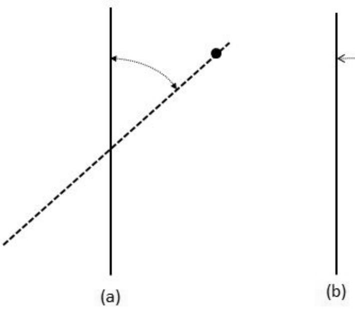


Fig. 1. (a) Hypothesis Margin, (b) Sample Margin

2.5 Decision Function

When a convergence is reached, we achieve the final prototypes $a_i = [u_{ik}]_{k=1,\ldots,l}$, $i = 1, \ldots, m$ and $b_j = [v_{jk}]_{k=1,\ldots,l}$, $j = 1, \ldots, n$.

For a new vector x, we can evaluate the distances from $\phi(x)$ to the prototypes using the following:

$$\begin{aligned}d(\phi(x), a_i) &= \|\phi(x) - a_i\|^2 = K(x,x) - 2\sum_{p=1}^{l} u_{ip}K(x_p, x) + \|a_i\|^2, \quad i = 1, \ldots, m\\ d(\phi(x), b_j) &= \|\phi(x) - b_j\|^2 = K(x,x) - 2\sum_{p=1}^{l} v_{ip}K(x_p, x) + \|b_j\|^2, \quad j = 1, \ldots, n\end{aligned} \tag{14}$$

The two closest prototypes to $\phi(x)$ and the decision function will be determined as follows

$$\begin{aligned}&i_0 = \arg\min_{1\le i\le m}\{d(\phi(x), a_i)\} \qquad j_0 = \arg\min_{1\le j\le n}\{d(\phi(x), b_j)\}\\ &f(x) = sign\left(d(\phi(x), b_{j_0}) - d(\phi(x), a_{i_0})\right)\end{aligned} \tag{15}$$

3 Experimental Results

The chosen data set was the well-known data set III provided by Department of Medical Informatics, Institute of Biomedical Engineering, Graz University of

Technology for motor imagery classification problem in BCI Competition II [9]. In data collection stage, a female normal subject was asked to sit in a relaxing chair with armrests and tried to control a feedback bar by means of imagery left or right hand movements. The sequences of left or right orders are random. The experiment consisted of 7 runs with 40 trials in each run. There were 280 trials in total and each of them lasted 9 seconds of which the first 3 seconds are used for preparation. Collected data was equally divided into two sets for training and testing. The data was recorded in three EEG channels which were $C3$, Cz and $C4$, sampled at $128Hz$, and filtered between 0.5 Hz and $30Hz$. Most of current algorithms only applied to the channels $C3$ and $C4$, and ignored the channel Cz. They argued that from brain theory, signals from channel Cz provide very little meaning to motor imagery problem. We truncated the first 3 seconds of each trial and used the rest for further processing. All trials are pre-processed by subtracting the ensemble mean of all trials. For each trial we extracted Combined Short-Window Bivariate Autoregressive Feature (CSWBVAR) parameters with window size of $512, 768$ data points corresponding to 1s-segment, 1.5s-segment and moving window step of $25\%, 50\%, 75\%$ of the window size. We did not try experiments with segment's size greater than $1.5s$ due to keeping signal approximately stationary and being comfortable with nature of brain signal.

The LVQ algorithms with different μ-functions mentioned above were performed in both the input and feature spaces to compare LVQ, GLVQ and HMLVQ with our proposed MLVQ (those are input space models), and to compare KLVQ, KGLVQ and KHMLVQ with our proposed KMLVQ (those are kernel feature space models). We also made comparison our method with Linear Support Vector Machine (LSVM) and Kernel Support Vector Machine (KSVM).

In our experiment, we did not use the sigmoid function $g(\mu, t) = \frac{1}{1+e^{-\mu t}}$ which results in the derivative $\frac{\partial g}{\partial \mu} = tg(1-g)$. Since the derivative of this function rapidly decreases to 0 when the time t approaches $+\infty$. For example when $t = 100$, the derivative is nearly equal to 0 if $-0.1 < \mu < 0.1$. Instead, we applied $g(\mu, t) = \frac{1}{1+e^{-\mu\sqrt{t}}}$ whose derivative is $\frac{\partial g}{\partial t} = \sqrt{t}g(1-g)$. This function shows two good features: 1) Its derivative approaches to 0 slower than that of the sigmoid function. 2) Given t, if $|\mu|$ of a vector exceeds a predefined threshold then the derivative or the rate at this vector is very small and the adaptation is minor.

The cross validation with 5 folds was employed. The learning rate α was set to 0.05. Both the number of positive and negative prototypes were set to 3. For Kernel LVQs, the popular RBF kernel function $K(x, x') = e^{-\gamma||x-x'||^2}$ was used. The parameter γ was searched in the grid $\{2^k : k = 2l, l = -4, -2, 0, 2, 4\}$. For KSVM, RBF kernel was applied, parameter γ varied in the grid $\{2^k : k = -15, -13, \ldots, 3\}$ and parameter C was searched in the grid $\{2^k : k = -15, -13, \ldots, 5\}$.

Experimental results are displayed in Table 1. The results indicate that for LVQs the classification accuracies of linear models and kernel models are not significantly different. The reason is that the data distributions in these data sets are not much complicated and some prototypes can be well-defined them. We emphasise in bold and bold-italic the cases where the proposed method outperforms others in the feature and input space, respectively. The results show that the proposed method outperforms others in both the input and feature space. KSVM is comparable to MLVQ and KMLVQ. However LVQs are simple, fast, intuitive and do not require running grid search over a massive range of parameters like SVMs. It points out the prospective for applying the proposed method to BCI data.

Table 1. Classification results (in %) on data set III (window size/ moving size) of BCI competition II for 5 space models LSVM, LVQ, GLVQ, HMLVQ and MLVQ, and 5 kernel feature space models KSVM, KLVQ, KGLVQ, KHMLVQ and KMLVQ

Data set	LSVM	LVQ	GLVQ	HMLVQ	MLVQ	KSVM	KLVQ	KGLVQ	KHMLVQ	KMLVQ
512/128	68%	70%	69%	72%	***77%***	77%	72%	73%	72%	**78%**
512/256	65%	72%	70%	68%	***74%***	75%	71%	71%	71%	**77%**
512/384	66%	69%	69%	70%	***75%***	**76%**	70%	71%	70%	**76%**
768/192	65%	71%	71%	69%	***73%***	73%	70%	71%	71%	**74%**
768/384	65%	70%	70%	69%	***72%***	**73%**	70%	71%	70%	**73%**
768/576	64%	69%	70%	68%	***73%***	**73%**	69%	71%	69%	**73%**

4 Conclusion

In this paper, we have introduced a new maximal margin approach to Kernel Generalised Learning Vector Quantisation which maximises the real margin which is crucial for kernel method. The new maximal margin approach can be applied to both the input space and feature space. The experiments performed on the well-known data set III of BCI competition II showed the prospective for applying this method to BCI data.

References

1. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Min. Knowl. Dis. 2, 121–167 (1998)
2. Cortes, C., Vapnik, V.: Support-Vector Networks. Mach. Learn., 273–297 (1995)
3. Crammer, K., Gilad-bachrach, R., Navot, A., Tishby, N.: Margin Analysis of the LVQ Algorithm. In: Advances in Neural Information Processing Systems 2002, pp. 462–469 (2002)
4. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. J. Comput. Syst. Sci. 55, 119–139 (1997)
5. Hammer, B., Villmann, T.: Generalized Relevance Learning Vector Quantization. Neural Netw. 15, 1059–1068 (2002)
6. Kohonen, T.: Self-organization and Associative Memory, 3rd edn. Springer, Berlin (1989)
7. Kohonen, T.: Learning Vector Quantization. In: The Handbook of Brain Theory and Neural Networks, pp. 537–540 (1995)
8. Liu, C.L., Nakagawa, M.: Evaluation of Prototype Learning Algorithms for Nearest-Neighbor Classifier in Application to Handwritten Character Recognition. Pattern Recogn. 34, 601–615 (2001)

9. Pfurtscheller, G., Schlögl, A.: Data Set III in BCI Competition II, http://www.bbci.de/competition/ii
10. Qinand, A.K., Suganthan, P.N.: A Novel Kernel Prototype-Based Learning Algorithm. In: 17th International Conference on Pattern Recognition, pp. 621–624 (2004)
11. Sato, A.: Discriminative Dimensionality Reduction Based on Generalized LVQ. In: Dorffner, G., Bischof, H., Hornik, K. (eds.) ICANN 2001. LNCS, vol. 2130, pp. 65–72. Springer, Heidelberg (2001)
12. Sato, A., Yamada, K.: Generalized Learning Vector Quantization. In: Neural Information Processing Systems Conference, pp. 423–429 (1995)
13. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1999)

A Basic Study on Particle Swarm Optimization Based on Chaotic Spike Oscillator Dynamics

Yoshikazu Yamanaka[1] and Tadashi Tsubone[2,*]

[1] Nagaoka University of Technology, 1603-1, Kamitomioka, Nagaoka, Niigata,
940-2188, Japan
y_yama@stn.nagaokaut.ac.jp
[2] Nagaoka University of Technology
tsubone@vos.nagaokaut.ac.jp

Abstract. In this paper, we propose a particle swarm optimization(abbr. PSO) based on chaotic spike oscillator dynamics(abbr. CSOPSO). Our method has ability to search optima without stochastic elements. Since the basic particle dynamics exhibits chaotic behavior on phase space consisting of the velocity and position, particles on the search space move with chaotic motion. Size of the chaotic attractor corresponding to search range of position can be controlled by single parameter. We focus on influence between size of the attractor and searching ability. The effectivity of CSOPSO by comparing with a previous PSO by some benchmark problems is considered.

Keywords: optimization problem, particle swarm optimization, chaos spiking oscillator.

1 Introduction

Particle Swarm Optimization (abbr. PSO) is a heuristic method for finding optimal solution of solution space based on the simulation of social behavior. PSO was developed by Kennedy and Eberhart in 1995[1],[2]. The algorithm searches a space by a population of individuals called particle. The particles are drawn toward the positions of previous best performance where is specified by its own and its companions flying experiences. Due to the simple concept and quick convergence, PSO is adapted to wide applications[5]. PSO shows efficient ability of balance between exploration and convergence because of combined stochastic elements. The elements effect the global search ability, however, it makes difficult to analyze behavior, convergence and stability.

Clerc and Kennedy proposed a simple deterministic PSO without stochastic elements[3]. Jin'no also proposed novel deterministic PSO with re-acceleration velocity scheme[4]. In this paper, we propose another optimization method without stochastic elements, PSO based on chaos spike oscillator dynamics(abbr. CSOPSO). In this method, the basic particle dynamics exhibits chaotic behavior on phase space consisting of particle velocity and position. Thereby particles on

[*] Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 199–207, 2012.

the search space move with chaotic motion. This method has a ability to search optima even though this method doesn't have any stochastic elements. Size of the chaotic attractor corresponding to search range of position is controlled by single parameter. We consider the influence between size of the attractor and its performance in Sec. 5.3 by comparing with a previous PSO[1],[2], the ability of searching solution is considered by some benchmark problems in Sec. 5.4.

2 Particle Swarm Optimization

The particle swarm optimization is an algorithm for finding an optimal solution using particles. The particles share the information of the best position and update their coordinates using personal and group experience. This algorithm was motived by social interaction[1].

In N-dimensional search space, a population of particles is initialized with random position x and velocity v. x and v is described as N-dimension vector. Every particles have own evaluated value calculated by function f, using the particle's positional coordinates. Positions, velocities and evaluated value are updated each time step. A particle keeps the own coordinate with the best evaluated value so far. The coordinate is defined by $pbest$. Likewise, all particles share the best coordinate as $gbest$.

The evaluated value of i-th particle which is located at x_i^t is compared with $f(pbest_i)$ and $f(gbest)$ each time step t. $pbest_i$ and $gbest$ is replaced by current position x_i^t, when x_i^t has better evaluated value than $f(pbest_i)$ and $f(gbest)$, respectively. At time step t, i-th particle updates its position and velocity according to follows [6]:

$$\boldsymbol{v_i^{t+1}} = \omega \boldsymbol{v_i^t} + c_1 Rand_1(\boldsymbol{pbest_i} - \boldsymbol{x_i^t}) + c_2 Rand_2(\boldsymbol{gbest} - \boldsymbol{x_i^t}), \tag{1}$$

$$\boldsymbol{x_i^{t+1}} = \boldsymbol{x_i^t} + \boldsymbol{v_i^{t+1}}, \tag{2}$$

where ω is inertia, c_1 and c_2 are positive constant, $Rand_1$ and $Rand_2$ are independently generated uniformed distribution with range [0 1].

3 PSO Based on Chaos Spike Oscillator Dynamics

This section describes about PSO based on Chaos Spike Oscillator Dynamics(CSOPSO). This method doesn't have any stochastic elements. The particles on the search space consisting of the particle velocity and position move with chaotic motion. The size of attractor is determined by single parameter and the size isn't depend on initial value.

In N dimensional search space, a population of particles is initialized with random position x described N-dimension vector. The velocity of this particle is v, N-dimension vector also. About i_{th} particle, the position and velocity is defined as follows:

$$\boldsymbol{x_i} = \left\{ \boldsymbol{x}_{(i,1)},\ x_{(i,2)},\ \ldots.\ ,\ x_{(i,N)} \right\}, \tag{3}$$

$$\boldsymbol{v_i} = \left\{ \boldsymbol{v}_{(i,1)},\ v_{(i,2)},\ \ldots.\ ,\ v_{(i,N)} \right\}. \tag{4}$$

In this method, the positional coordinates of the particles are evaluated by object function f. All particles update the best position as *pbest* if necessary. Global best position, *gbest*, is selected from the best position of *pbest*s among population. The particles update their position and velocity according to Chaos Spike Oscillator Dynamics(CSO). The unstable fixed point of this dynamics is described by *pbest* and *gbest*. Positions and velocities are updated using expanding and rotation on the fixed point. This method is described more details in following paragraphs.

3.1 Updating Particle Position and Velocity

The particles are updated by 2 operation, 1) rotation and expansion, 2)jumping toward an unstable fixed pint. In phase space consisting of particle velocity and position, basically, the particles expand and rotate around an unstable fixed point. The unstable fixed point is described in Sec. 3.3. Once a particle satisfies a condition about position and velocity, the particle jumps toward the unstable fixed point. By these 2 operation, the trajectory of particle exhibit chaotic behavior on the phase space.

Dynamics on Phase Space. First, we focus on the phase space consisting of particle velocity and position. To be easy to describe, linear transformed position $y_{(i,j)}$ of the i-th particle on j-th dimension is defined by

$$y_{(i,j)} = x_{(i,j)} - fp_{(i,j)}, \tag{5}$$

where $x_{(i,j)}$ is a position of a particle in real-searching field. The fixed point, $fp_{(i,j)}$, was described in Sec. 3.3. The fixed point is set to the origin on $v - y$ phase space. At time-step t, i-th particle position on j-th dimension is updated by the following dynamics.

$$\begin{bmatrix} y_{(i,j,t+1)} \\ v_{(i,j,t+1)} \end{bmatrix} = \begin{cases} \begin{bmatrix} 2y_{th1(i,j)} - y_{(i,j,t)} \\ 0 \end{bmatrix} & \text{(6a)} \\ \quad \text{for } (y_{(i,j,t)} < y_{th1(i,j)}) \text{ and } (v_{(i,j,t)} \geq 0) & \text{(6b)} \\ \begin{bmatrix} 2y_{th2(i,j)} - y_{(i,j,t)} \\ 0 \end{bmatrix} & \text{(6c)} \\ \quad \text{for } (y_{(i,j,t)} > y_{th2(i,j)}) \text{ and } (v_{(i,j,t)} = 0) & \text{(6d)} \\ R \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} y_{(i,j,t)} \\ v_{(i,j,t)} \end{bmatrix} \quad \text{otherwise,} & \text{(6e)} \end{cases}$$

where $y_{th1}(< 0)$ is a threshold parameter, y_{th2} is the intersection of limit cycle and $v = 0$, R is a damping parameter and θ is a degree parameter.

We define y_{th2} such that located on the intersection of limit cycle and $v = 0$. y_{th2} is given by

$$y_{th2(i,j)} = \frac{2y_{th1(i,j)}}{1 - R^{\frac{\pi}{\theta}}}. \tag{7}$$

Once particles satisfy the condition (6b) and (6d), the position of particle jumps toward the fixed point by (6a) and (6c) respactively. By this dynamics, particles exhibit chaotic motion in the range adjusted by y_{th1} not depending on initial position and velocity.

Example. A particle trajectory is shown in Fig. 1.

First, a particle is initialized $y = 5, v = 0.01$ shown as P0. The particle is moved to P1 following (6e). At P1, the particle satisfies (6b), then the particle jumps to P2 by (6a). At P2, the particle satisfies (6d), then it re-jumpes to P3 following (6c). The particle is moved from P3 to P4, and jumps to P5. At P5, the particle doesn't satisfy (6d), then it begins to be moved following (6e). This dynamics repeats this manner and exhibits chaotic attractor as shown in Fig. 2.

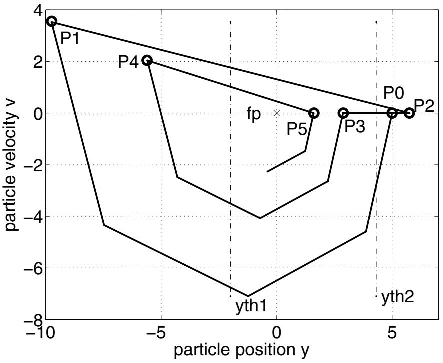


Fig. 1. A particle trajectory on y-v phase space. $\theta = 0.87$ (50[deg]), R = 1.2, $yth_1 = -2$, initialize(y, v)=(5, 0.01), 10 iteration.

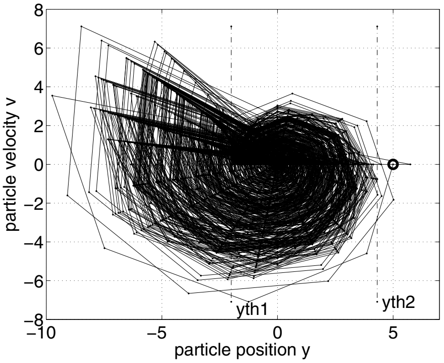


Fig. 2. A particle trajectory on y-v phase space. $\theta = 0.87$ (50[deg]), R = 1.2, $yth_1 = -2$, initialize(y, v)=(5, 0.01), 4000 iteration.

3.2 Evaluation of Particles

An object function f is defined. Each time step, current position of particles are evaluated by f. All particles keep the best coordinate as *pbest*, and share the best coordinate in all particles as *gbest*. When a particle finds better position than *pbest* and *gbest*, these best position is updated by the new-best position, respectively.

3.3 Dynamics of Fixed Point

In this paragraph, the fixed point fp is described. The fixed point is defined independently. The number of fixed point of a particle is same amount of the number of dimensions. In N dimensional searching space, the fixed point $\boldsymbol{fp_i}$ of i-th particle is described by

$$\boldsymbol{fp_i} = \left\{ fp_{(i,1)},\ fp_{(i,2)},\ \dots\ ,\ fp_{(i,N)} \right\}. \tag{8}$$

In the phase space consisting of particle position and velocity, fp is on $v = 0$. The fp is updated by *pbest* and *gbest* each time step. At time step t, the next fixed point of i-th particle, on j-th dimension is defined as follows:

$$fp_{(i,j,t+1)} = fp_{(i,j,t)} + c1(pbest_{(i,j,t)} - fp_{(i,j,t)}) + c2(gbest_{(j,t)} - fp_{(i,j,t)}), \quad (9)$$

where $c1$ and $c2$ are positive constants.

Updating fixed point, each particle searches optimal position around *pbest* and *gbest* with chaotic behavior.

3.4 Algorithm of CSOPSO

The basic algorithm of CSOPSO in pseudocode follows.

```
Initialize Population
Initialize $pbest$ and $gbest$
Do
    Evaluate all Populations
    For $i = 1$ to Population Size
        For $j$ = 1 to Dimension
            update $fp_{ij}$
            $y_{ij} = x_{ij} - fp_{ij}$
            if $(v_{ij} \geq 0)$ & $(y_{ij} < y_{th1_{ij}})$ then
                $v_{ij} = 0$
                $y_{ij} = 2y_{th1_{ij}} - y_{ij}$
            else if $(v_{ij} = 0)$ & $(y_{ij} > y_{th2_{ij}})$ then
                $y_{ij} = 2y_{th2_{ij}} - y_{ij}$
            else
            $\begin{bmatrix} y_{ij} \\ v_{ij} \end{bmatrix} = R \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} y_{ij} \\ v_{ij} \end{bmatrix}$
            end if
            $x_{ij} = y_{ij} + fp_{ij}$
        next $j$
    next $i$
Until termination criterion is met
```

4 Reducing Size of chaotic attractor

It is well known that the balance between global and local search is effective for optimizer[6]. In proposed method, the size of chaotic attractor is identified by y_{th1}. The size of attractor provides the searching regions. This feature has a possibility to change searching region each time step. We apply reducing searching region by decreasing y_{th1} linearly.

In this article, two linear reducing y_{th1} models are proposed. Reducing y_{th1} by (10) and (11) is used by CSOPSO1 and CSOPSO2, respectively.

$$y_{th1} = \frac{y_{th1_{final}} - y_{th1_{initial}}}{t_{max}} t + y_{th1_{initial}}, \quad (10)$$

$$y_{th1} = y_{th1_{initial}} e^{-\frac{a}{t_{max}} t}, \tag{11}$$

where $y_{th1_{initial}}$ is initial value of y_{th1}, $y_{th_{final}}$ is final value of y_{th1} at the last iteration, a is a positive constant and t_{max} is a number of iteration.

For comparison, the basic algorithm of CSOPSO, without reducing y_{th1} model, is represented by CSOPSObasic.

5 Experimental Results

In order to compare the performance, proposed models are applied to some benchmark functions. In this section, we compare the influence of attractor size in Sec. 5.3 through three models, CSOPSObasic, CSOPSO1 and CSOPSO2 for Rastrigin function. We also compare the performance between proposed models and PSO through four benchmark functions in Sec. 5.4.

5.1 Parameter Selection for CSOPSO models

For comparison, the parameters used in proposed models are selected experimentally. For these models, we suppose $c1 = c2 = c$. The common parameters in proposed models are set to $\theta = 0.367$, $R = 1.05$ and $c = 0.01$ by simulating with some patterns of parameters. The population size is set to 30 for all models.

5.2 Benchmark Functions

For comparison four non-linear benchmark functions are described in Table 1. The initial positional range for these functions are set to $[-10\ 10]$.

Table 1. Benchmark functions

f	Function	Domain	Minimum Value
Sphere	$f_0(x) = \sum_{i=1}^{N} x_i^2$	$[-10\ 10]^N$	$f_0(0) = 0$
Rastrigin	$f_1(x) = \sum_{i=1}^{N}(x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-10\ 10]^N$	$f_1(0) = 0$
Rosenbrock	$f_2(x) = \sum_{i=1}^{N-1} 100\left((x_{i+1}^2 - x_i^2)^2 + (1 - x_i)^2\right)$	$[-10\ 10]^N$	$f_2(1) = 0$
Griewank	$f_3(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod\cos(\frac{x_i}{\sqrt{i}})$	$[-10\ 10]^N$	$f_3(0) = 0$

5.3 The Influence of Attractor Size for Performance

We consider the influence on attractor size for performance. In this section, proposed models are adapted to 30 dimensional Rastrigin function f_1. Initialized range of the particles position is set $[-10\ 10]$ with uniform distribution. Initialized velocity is set to 0.

The average best function values of CSOPSObasic model under 30 independent runs each max iteration with some y_{th1} are shown in Fig. 3. Figure 3 shows that the performance is depending on the size of attractors.

The average best function values of CSOPSO1 and CSOPSO2 under 30 independent runs each max iteration are shown in Fig. 4 and Fig. 5, respectively. In these simulations, both models reduce y_{th1} from -4.0 to different final y_{th1} values. Figure 4 shows that the performance of CSOPSO1 is worse than CSOPSObasic. In this simulation, the attractor size is bigger than CSOPSObasic in Fig. 3 during almost time step. However, CSOPSO2 shows better performance than CSOPSO1 even though initial y_{th} is same value as CSOPSO1. It is seems that CSOPSO2 has the better balances between local search ability and global search ability.

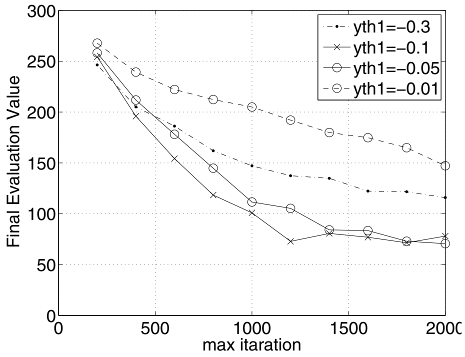


Fig. 3. Comparison of y_{th1} with CSOPSObasic

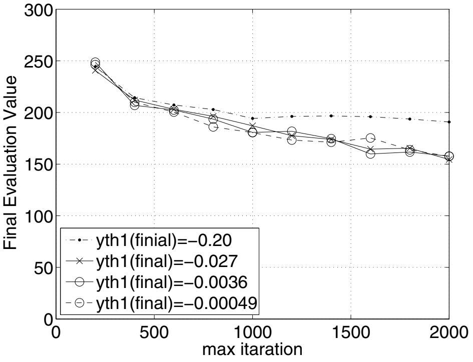


Fig. 4. Comparison of y_{th1} with CSOPSO1

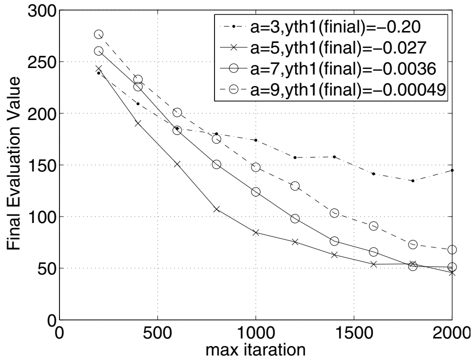


Fig. 5. Comparison of 'a' with CSOPSO2

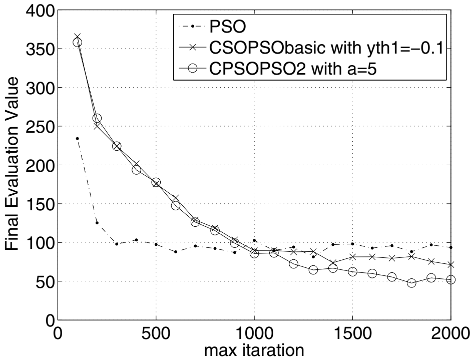


Fig. 6. Comparison of Algorithms

5.4 Comaprison CSOPSO with PSO

Figure 6 shows the average best function values of PSO, CSOPSObasic and CSOPSO2 for 30 dimensional Rastrigin function under 30 independent runs each max iteration. The parameters for PSO algorithm described by (1) and (2) are followings, $\omega = 0.729$ and $c1 = c2 = 1.49445$. These parameters were introduced in [7]. The parameters for proposed models are same as described in Sec. 5.1. Population size of 30 is used in PSO and proposed models.

In Fig. 6 it seems that proposed model, CSOPSObasic and CSOPSO2, performs better than PSO over 1500 iteration.

Table 2 shows the average best function value and standard deviation with bracket among PSO, CSOPSObasic and CSOPSO2 for the benchmark functions: Sphere, Rastrigin, Rosenbrock and Griewank under 30 independent runs. For PSO parameters, $\omega = 0.729$ and $c1 = c2 = 1.49445$ [7], are adapted. For CSOPSO models, the common parameters are set to $\theta = 0.367$, $R = 1.05$ and $c1 = c2 = 0.01$. For CSOPSObasic, y_{th1} is set to 0.01. For CSOPSO2, initial $y_{th1} = -4$, $a = 5$ is adapted. The proposed model performs well for Rastrigin function f_1 at 2000 and 3000 iteration. It seems that CSOPSO2 model shows better performance for Griewank function f_3 at 2000 and 3000 iteration and almost equal performance for Rosenbrock function f_2 at 2000 and 3000 iteration.

Table 2. Comparison with PSO and CSOPSO

f	Dimension	iteration	PSO	CSOPSObasic	CSOPSO2
f_0	30	1000	0.000 (0.0000)	3.961 (2.26)	0.1745 (0.22)
f_0	30	2000	0.000 (0.0000)	0.5499 (1.13)	0.004975 (0.000901)
f_0	30	3000	0.000 (0.0000)	0.03899 (0.0741)	0.004471 (0.000667)
f_1	30	1000	88.15 (26.4)	93.05 (23.3)	92.03 (23.7)
f_1	30	2000	88.42 (26.4)	76.13 (25.2)	49.62 (17.9)
f_1	30	3000	78.57 (23.7)	70.92 (18.5)	48.57 (11.4)
f_2	30	1000	35.74 (28.6)	650.0 (347)	111.3 (98.7)
f_2	30	2000	41.12 (35.5)	104.0 (80.6)	34.47 (28.6)
f_2	30	3000	26.67 (29.8)	37.97 (16.5)	32.03 (16.5)
f_3	30	1000	0.009927 (0.0105)	0.2638 (0.130)	0.06420 (0.0440)
f_3	30	2000	0.006566 (0.00926)	0.05045 (0.0586)	0.004468 (0.0593)
f_3	30	3000	0.007468 (0.00868)	0.005834 (0.00671)	0.00558 (0.0957)

6 Conclusion

In this paper we proposed PSO based on chaos spike oscillator dynamics(CSOPSO). Nevertheless our proposed method doesn't have any stochastic elements and the parameters are not tuned enough, it exhibited better performance for Rastrigin and Griwank functions at 3000 iteration compared with PSO. These results suggest that CSOPSO performance can be improved. In the future, we will consider about providing other models to decrease size of attractor and studying about influence of performance depending on each parameters.

References

1. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proc. IEEE Int. Conf. Neural Networks, pp. 1942-1948 (1995)
2. Eberhart, R., Kennedy, J.: A New Optimizer Using Particle Swarm Theory. In: Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39-43 (1995)
3. Clerc, M., Kennedy, J.: The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE Trans. Evol. Comput. 6, 58-73 (2002)
4. Jin'no K.:, A Novel Deterministic Particle Swarm Optimization System. Journal of Signal Processing. 13, 507-513 (2009)
5. AlRashidi, M.R., El-Hawary, M.E.: A Survey of Particle Swarm Optimization Applications in Electric Power Systems. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. 13, 913-918 (2009)
6. Shi Y., Eberhart, R.: A Modified Particle Swarm Optimizer. In: Proc. of the IEEE Congress on Evolutionary Computation, pp. 69-73. IEEE Service Center, USA (1998)
7. Eberhart, R.C., Shi, Y.: Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. In: Proc. 2000 Congr. Evolutionary Computation, pp. 84-88. San Diego, CA (2000)

On the Objective Function and Learning Algorithm for Concurrent Open Node Fault

Chi Sing Leung[1],*, Pui Fai Sum[2], and Kai-Tat Ng[1]

[1] Dept. of Electronic Engineering, City University of Hong Kong, Hong Kong
eeleungc@cityu.edu.hk

[2] Institute of Technology Management, National Chung Hsing University

Abstract. This paper studies the performance of faulty RBF networks when stuck-at-zero node fault and stuck-at-one node fault happen. An objective function for training fault tolerant RBF networks for node fault is first derived. A training learning algorithm for faulty RBF networks is then presented. Finally, a mean prediction error formula which can estimate the test set error of faulty networks is derived. Simulation experiments are then performed to verify our theoretical result.

Keywords: Fault tolerance, RBF networks, generalization ability.

1 Introduction

Regularization [1] is an effective technique to train a neural network with good generalization. In this technique, the usual assumption is that trained neural networks can be perfectly implemented. However, for electronic implementations, many network fault situations [2], [3], such as component failure, sign bit change, and open circuit, could happen. If special care is not taken, the performance of a neural network could degrade drastically when network fault appears [4–6]. Hence, obtaining a fault tolerant neural network is very important.

In the implementation of neural networks, node fault, such as stuck-at-zero and stuck-at-one, happens unavoidably. The classical way to improve the fault tolerance is to generate a number of faulty networks during training [4]. But the number of training iterations should be very large. Otherwise, the learning algorithm cannot capture the statistical behavior of network fault. In [6], the fault tolerant problem was formulated as an unconstrained optimization problem. Those formulation can improve the fault tolerance of faulty networks. However, they are computationally complicated when the multi-node fault situation and multi-fault model are considered, because the number of potential faulty networks for the multi-node fault situation and multi-fault model is very large. Besides, although many fault tolerant training methods were developed in the past three decades, most of them focused on one kind of node faults. For example, in [5], [6], the algorithm was used to handle the stuck-at-zero only.

This paper investigates how the node fault situation affects the performance of RBF networks when the stuck-at-zero and stuck-at-one concurrently happens. We then derive an objective function for this concurrent situation. The corresponding training algorithm is then developed.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 208–216, 2012.

2 Background

Consider that we are given a training set: $\mathcal{D}_t = \{(\boldsymbol{x}_i, y_i) : \boldsymbol{x}_i \in \Re^K, y_i \in \Re, i = 1, \cdots, N\}$, where $\boldsymbol{x}_i$ and y_i are the input and output of the i-th sample, respectively, and K is the input dimension. The output is generated by an unknown stochastic system, given by $y_i = f(\boldsymbol{x}_i) + \epsilon_i$, where $f(\cdot)$ is a nonlinear function, and ϵ_i's are the independent zero-mean Gaussian random variables with variance σ_ϵ^2. In the RBF approach, the unknown system $f(\cdot)$ is approximated by

$$f(\boldsymbol{x}) \approx \hat{f}(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=1}^{M} w_j \phi_j(\boldsymbol{x}) = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{w} \tag{1}$$

where w_j's are weights, $\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \cdots, \phi_M(\boldsymbol{x})]^T$, and $\phi_j(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}_j\|^2}{\Delta}\right)$ is the j-th basis function. $\boldsymbol{c}_j$'s are the RBF centers. Parameter Δ controls the width of RBF kernels. The training set error $\mathcal{E}(\mathcal{D}_t)$ is given by

$$\mathcal{E}(\mathcal{D}_t) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \boldsymbol{\phi}^T(\boldsymbol{x}_i)\boldsymbol{w})^2. \tag{2}$$

Among different forms of node faults, stuck-at-zero and stuck-at-one are the most common fault models [5–8]. In stuck-at-zero, the output of a node is tied to zero. In stuck-at-one, the output of a node is tied to one. We propose a general model to describe the co-existing of stuck-at-zero and stuck-at-one, given by

$$\tilde{\phi}_j(\boldsymbol{x}) = (1 - \beta_j^2)\phi_j(\boldsymbol{x}) + \frac{1}{2}\beta_j(1 + \beta_j), \quad \forall \ j = 1, \cdots, M. \tag{3}$$

In our formulation, β_j's are fault factors which describe the fault situation of the RBF nodes, given by

$$\beta_j = \begin{cases} 0 & \text{no fault} \\ 1 & \text{stuck-at-one} \\ -1 & \text{stuck-at-zero} \end{cases}. \tag{4}$$

If the jth node ($\beta_j = 0$) is no fault, then $\tilde{\phi}_j(\boldsymbol{x}) = \phi_j(\boldsymbol{x})$. If it is stuck-at-one ($\beta_j = 1$), then $\tilde{\phi}_j(\boldsymbol{x}) = 1$. If it is stuck-at-zero ($\beta_j = 1$), then $\tilde{\phi}_j(\boldsymbol{x}) = 0$. The probability mass function of the fault factor is given by Prob$(\beta_j = 1) = p_1$, Prob$(\beta_j = -1) = p_o$, and Prob$(\beta_j = 0) = 1 - p_1 - p_o$.

3 Performance, Objective Function, and Training Algorithm

With the proposed fault model and fault statistics, we can study the performance of faulty networks. Given the fault factors β_j's, from (3), the faulty training error is

$$\mathcal{E}(\mathcal{D}_t)_\beta = \frac{1}{N} \sum_{i=1}^{N} (y_i - \tilde{\boldsymbol{\phi}}^T(\boldsymbol{x}_i)\boldsymbol{w})^2 \tag{5}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left\{ y_i^2 - 2y_i \sum_{j=1}^{M} w_j \tilde{\phi}_j(\boldsymbol{x}_i) + \sum_{j=1}^{M} \sum_{j' \neq j}^{M} w_j w_{j'} \tilde{\phi}_j(\boldsymbol{x}_i) \tilde{\phi}_{j'}(\boldsymbol{x}_i) + \sum_{j=1}^{M} w_j^2 \tilde{\phi}_j^2(\boldsymbol{x}_i) \right\} \tag{6}$$

where $\tilde{\boldsymbol{\phi}}(\boldsymbol{x}_i) = [\tilde{\phi}_1(\boldsymbol{x}_i), \cdots, \tilde{\phi}_M(\boldsymbol{x}_i)]^T$. According to the definition of the node fault,

$$\langle \tilde{\phi}_j(\boldsymbol{x}_i) \rangle = (1 - p_1 - p_o)\phi_j(\boldsymbol{x}_i) + p_1 \tag{7a}$$

$$\langle \tilde{\phi}_j^2(\boldsymbol{x}_i) \rangle = (1 - p_1 - p_o)\phi_j^2(\boldsymbol{x}_i) + p_1 \tag{7b}$$

$$\langle \tilde{\phi}_j(\boldsymbol{x}_i)\tilde{\phi}_{j'}(\boldsymbol{x}_i) \rangle = [(1 - p_1 - p_o)\phi_j(\boldsymbol{x}_i) + p_1][(1 - p_1 - p_o)\phi_{j'}(\boldsymbol{x}_i) + p_1] \tag{7c}$$

where $\langle \cdot \rangle$ is the expectation operation over the fault factors. Based on (7), the training error of a faulty network is given by

$$\begin{aligned}
\bar{\mathcal{E}}(\mathcal{D}_t)_\beta = \frac{1}{N}\sum_{i=1}^{N} \Bigg\{ & y_i^2 - 2(1 - p_1 - p_o) y_i \sum_{j=1}^{M} w_j \phi_j(\boldsymbol{x}_i) - 2p_1 y_i \sum_{j=1}^{M} w_j \\
& + (1 - p_1 - p_o)^2 \sum_{j=1}^{M} \sum_{j' \neq j}^{M} w_j w_{j'} \phi_j(\boldsymbol{x}_i)\phi_{j'}(\boldsymbol{x}_i) \\
& + p_1(1 - p_1 - p_o) \sum_{j=1}^{M} \sum_{j' \neq j}^{M} w_j w_{j'} [\phi_j(\boldsymbol{x}_i) + \phi_{j'}(\boldsymbol{x}_i)] + p_1^2 \sum_{j=1}^{M} \sum_{j' \neq j}^{M} w_j w_{j'} \\
& + (1 - p_1 - p_o) \sum_{j=1}^{M} w_j^2 \phi_j^2(\boldsymbol{x}_i) + p_1 \sum_{j=1}^{M} w_j^2 \Bigg\} .
\end{aligned} \tag{8}$$

From the fact that $\sum_{j=1}^{M} \sum_{j' \neq j}^{M} a_j a_{j'} = \sum_{j=1}^{M} \sum_{j'=1}^{M} a_j a_{j'} - \sum_{j=1}^{M} a_j^2$, (8) becomes

$$\begin{aligned}
\bar{\mathcal{E}}(\mathcal{D}_t)_\beta = & \frac{p_1 + p_o}{N} \sum_{i=1}^{N} y_i^2 + (1 - p_1 - p_o) \frac{1}{N} \sum_{i=1}^{N} (y_i - \boldsymbol{\phi}^T(\boldsymbol{x}_i)\boldsymbol{w})^2 - \frac{2p_1}{N} \sum_{i=1}^{N} y_i \mathbf{1}^T \boldsymbol{w} \\
& + (1 - p_1 - p_o)(p_1 + p_o)\boldsymbol{w}^T(\boldsymbol{G} - \boldsymbol{H})\boldsymbol{w} - 2p_1(1 - p_1 - p_o)\boldsymbol{w}^T \boldsymbol{D} \boldsymbol{w} \\
& + p_1(1 - p_1)\boldsymbol{w}^T \boldsymbol{w} + p_1(1 - p_1 - p_o)\boldsymbol{w}^T \boldsymbol{\Theta} \boldsymbol{w} + p_1^2 \boldsymbol{w}^T \underline{\mathbf{1}} \boldsymbol{w}
\end{aligned} \tag{9}$$

where $\boldsymbol{H} = \frac{1}{N} \sum_{j=1}^{N} \boldsymbol{\phi}(\boldsymbol{x}_i)\boldsymbol{\phi}^T(\boldsymbol{x}_i)$, $\boldsymbol{G} = \mathbf{diag}(\boldsymbol{H})$, $\mathbf{1} = [1, \cdots, 1]^T$,

$$\boldsymbol{\Theta} = \frac{1}{N}\sum_{i=1}^{N} \begin{pmatrix} \phi_1(\boldsymbol{x}_i) & \phi_2(\boldsymbol{x}_i) & \cdots & \phi_M(\boldsymbol{x}_i) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1(\boldsymbol{x}_i) & \phi_2(\boldsymbol{x}_i) & \cdots & \phi_M(\boldsymbol{x}_i) \\ \phi_1(\boldsymbol{x}_i) & \phi_2(\boldsymbol{x}_i) & \cdots & \phi_M(\boldsymbol{x}_i) \end{pmatrix} + \begin{pmatrix} \phi_1(\boldsymbol{x}_i) & \cdots & \phi_1(\boldsymbol{x}_i) & \phi_1(\boldsymbol{x}_i) \\ \phi_2(\boldsymbol{x}_i) & \cdots & \phi_2(\boldsymbol{x}_i) & \phi_2(\boldsymbol{x}_i) \\ \vdots & \cdots & \vdots & \vdots \\ \phi_M(\boldsymbol{x}_i) & \cdots & \phi_M(\boldsymbol{x}_i) & \phi_M(\boldsymbol{x}_i) \end{pmatrix} \tag{10}$$

$$\boldsymbol{D} = \frac{1}{N}\sum_{i=1}^{N} \begin{pmatrix} \phi_1(\boldsymbol{x}_i) & 0 & \cdots & 0 \\ 0 & \phi_2(\boldsymbol{x}_i) & 0 & \cdots \\ \vdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & \phi_M(\boldsymbol{x}_i) \end{pmatrix}, \text{ and } \underline{\mathbf{1}} = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} . \tag{11}$$

Equation (9) tells us the training error of faulty networks. Since the term $\frac{p_1+p_o}{N}\sum_{j=1}^{N} y_i^2$ in (9) is independent of $\boldsymbol{w}$, minimizing the training error of faulty networks is equivalent to minimizing *the following objective function*:

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}) = &(1-p_1-p_o)\frac{1}{N}\sum_{i=1}^{N}(y_i-\boldsymbol{\phi}^T(\boldsymbol{x}_i)\boldsymbol{w})^2-\frac{2p_1}{N}\sum_{i=1}^{N}y_i\mathbf{1}^T\boldsymbol{w}\\ &+(1-p_1-p_o)(p_1+p_o)\boldsymbol{w}^T(\boldsymbol{G}-\boldsymbol{H})\boldsymbol{w}-2p_1(1-p_1-p_o)\boldsymbol{w}^T\boldsymbol{D}\boldsymbol{w}\\ &+p_1(1-p_1)\boldsymbol{w}^T\boldsymbol{w}+p_1(1-p_1-p_o)\boldsymbol{w}^T\boldsymbol{\Theta}\boldsymbol{w}+p_1^2\boldsymbol{w}^T\underline{\mathbf{1}}\boldsymbol{w}\end{aligned} \tag{12}$$

In (12), the first term corresponds to the training error of a fault-free network and other terms are similar to the conventional penalty term.

The optimal weight vector for minimizing the objective function can be obtained by considering the derivative of $\mathcal{L}(\boldsymbol{w})$ respect to $\boldsymbol{w}$. It is given by

$$\boldsymbol{w} = \boldsymbol{\Gamma}^{-1}\frac{1}{N}\sum_{i=1}^{N}((1-p_1-p_o)\boldsymbol{\phi}(\boldsymbol{x}_i)+p_1\mathbf{1})\cdot y_i\,, \tag{13}$$

$$\begin{aligned}\boldsymbol{\Gamma} = &\{(1-p_1-p_o)(H+(p_1+p_o)(\boldsymbol{G}-\boldsymbol{H})-2p_1\boldsymbol{D}+p_1\boldsymbol{\Theta})\\ &+p_1(1-p_1)\boldsymbol{I}+p_1^2\underline{\mathbf{1}}\}\,,\end{aligned} \tag{14}$$

and $\boldsymbol{I}$ is an identity matrix.

4 Estimation of Generalization Error

With (13), we know how to train a fault tolerant RBF network for concurrent node fault, where stuck-at-zero and stuck-at-one appear at the same time. However, in many situations, we would like to know how well the network performs on unseen samples. This section derives a mean prediction error (MPE) formula to estimate the generalization ability for faulty networks trained with (13). The training error can also be expressed as

$$\begin{aligned}\bar{\mathcal{E}}(\mathcal{D}_t)_\beta = &\langle y^2\rangle_{\mathcal{D}_t}-2(1-p_1-p_o)\langle y\boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{w}\rangle_{\mathcal{D}_t}+(1-p_1-p_o)^2\boldsymbol{w}^T\boldsymbol{H}\boldsymbol{w}-2p_1\langle y\mathbf{1}^T\boldsymbol{w}\rangle_{\mathcal{D}_t}\\ &+(1-p_1-p_o)(p_1+p_o)\boldsymbol{w}^T\boldsymbol{G}w-2p_1(1-p_1-p_o)\boldsymbol{w}^T\boldsymbol{D}\boldsymbol{w}\\ &+p_1(1-p_1)\boldsymbol{w}^T\boldsymbol{w}+p_1(1-p_1-p_o)\boldsymbol{w}^T\boldsymbol{\Theta}\boldsymbol{w}+p_1^2\boldsymbol{w}^T\underline{\mathbf{1}}\boldsymbol{w}.\end{aligned} \tag{15}$$

Similarly, for the test set $\mathcal{D}_f=\{(\boldsymbol{x}'_{i'},y'_{i'})_{i=1}^{N'}$, the error of faulty networks is given by

$$\begin{aligned}\bar{\mathcal{E}}(\mathcal{D}_f)_\beta = &\langle y'^2\rangle_{\mathcal{D}_f}-2(1-p_1-p_o)\langle y'\boldsymbol{\phi}^T(\boldsymbol{x}')\boldsymbol{w}\rangle_{\mathcal{D}_f}+(1-p_1-p_o)^2\boldsymbol{w}^T\boldsymbol{H}'\boldsymbol{w}-2p_1\langle y'\mathbf{1}^T\boldsymbol{w}\rangle_{\mathcal{D}_f}\\ &+(1-p_1-p_o)(p_1+p_o)\boldsymbol{w}^T\boldsymbol{G}'w-2p_1(1-p_1-p_o)\boldsymbol{w}^T\boldsymbol{D}'\boldsymbol{w}\\ &+p_1(1-p_1)\boldsymbol{w}^T\boldsymbol{w}+p_1(1-p_1-p_o)\boldsymbol{w}^T\boldsymbol{\Theta}'\boldsymbol{w}+p_1^2\boldsymbol{w}^T\underline{\mathbf{1}}\boldsymbol{w}.\end{aligned} \tag{16}$$

where

$$\boldsymbol{\Theta}'=\frac{1}{N'}\sum_{i'=1}^{N'}\begin{pmatrix}\phi_1(\boldsymbol{x}'_{i'}) & \phi_2(\boldsymbol{x}'_{i'}) & \cdots & \phi_M(\boldsymbol{x}'_{i'})\\ \vdots & \vdots & \vdots & \vdots\\ \phi_1(\boldsymbol{x}'_{i'}) & \phi_2(\boldsymbol{x}'_{i'}) & \cdots & \phi_M(\boldsymbol{x}'_{i'})\\ \phi_1(\boldsymbol{x}'_{i'}) & \phi_2(\boldsymbol{x}'_{i'}) & \cdots & \phi_M(\boldsymbol{x}'_{i'})\end{pmatrix}+\begin{pmatrix}\phi_1(\boldsymbol{x}'_{i'}) & \cdots & \phi_1(\boldsymbol{x}'_{i'}) & \phi_1(\boldsymbol{x}'_{i'})\\ \phi_2(\boldsymbol{x}'_{i'}) & \cdots & \phi_2(\boldsymbol{x}'_{i'}) & \phi_2(\boldsymbol{x}'_{i'})\\ \vdots & \cdots & \vdots & \vdots\\ \phi_M(\boldsymbol{x}'_{i'}) & \cdots & \phi_M(\boldsymbol{x}'_{i'}) & \phi_M(\boldsymbol{x}'_{i'})\end{pmatrix}, \tag{17}$$

$\boldsymbol{H}'=\frac{1}{N'}\sum_{i'=1}^{N'}\boldsymbol{\phi}(\boldsymbol{x}'_{i'})\boldsymbol{\phi}^T(\boldsymbol{x}'_{i'})$, $\boldsymbol{G}'=\textbf{diag}(\boldsymbol{H}')$, and $\boldsymbol{D}'=\frac{1}{2}\textbf{diag}(\boldsymbol{\Theta}')$.

Denote the true weight vector as $\boldsymbol{w}_o$. Hence,

$$y_i = \boldsymbol{\phi}^T(\boldsymbol{x}_i)\boldsymbol{w}_o + \epsilon_i \quad \text{and} \quad y'_{i'} = \boldsymbol{\phi}^T(\boldsymbol{x}'_{i'})\boldsymbol{w}_o + \epsilon'_{i'}, \tag{18}$$

where ϵ_i's and $\epsilon'_{i'}$'s are independent zero-mean Gaussian random variables with variance σ_ϵ^2. From (13) and (18), the term $\langle y\boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{w}\rangle_{\mathcal{D}_t}$ in (15) is given by

$$\left\langle [\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{w}_o^T\boldsymbol{\phi}(\boldsymbol{w}_i)+\epsilon_i)\boldsymbol{\phi}^T(\boldsymbol{x}_i)]\boldsymbol{\Gamma}^{-1}[\frac{1}{N}\sum_{i=1}^{N}((1-p_1-p_o)\boldsymbol{\phi}(\boldsymbol{x}_i)+p_1\mathbf{1})(\boldsymbol{w}_o^T\boldsymbol{\phi}(\boldsymbol{w}_i)+\epsilon_i)]\right\rangle_{\epsilon_i} . \tag{19}$$

Since ϵ_i's are independent,

$$\begin{aligned}\langle y\boldsymbol{\phi}^T(\boldsymbol{x})\rangle_{\mathcal{D}_t} &= \boldsymbol{w}_o^T\boldsymbol{H}\boldsymbol{\Gamma}^{-1}[(1-p_1-p_o)\boldsymbol{H}+p_1\boldsymbol{\Phi}]\boldsymbol{w}_o \\ &\quad + \frac{(1-p_1-p_o)\sigma_\epsilon^2}{N}\mathrm{Tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{H}) + \frac{p_1\sigma_\epsilon^2}{N}\overline{\boldsymbol{\phi}}^T\boldsymbol{\Gamma}^{-1}\mathbf{1}\,,\end{aligned} \tag{20}$$

where $\overline{\boldsymbol{\phi}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_i)$,

$\boldsymbol{\Phi} = \frac{1}{N}\sum_{i=1}^{N}\begin{pmatrix}\phi_1(\boldsymbol{x}_i) & \phi_2(\boldsymbol{x}_i) & \cdots & \phi_M(\boldsymbol{x}_i)\\ \vdots & \vdots & \vdots & \vdots\\ \phi_1(\boldsymbol{x}_i) & \phi_2(\boldsymbol{x}_i) & \cdots & \phi_M(\boldsymbol{x}_i)\\ \phi_1(\boldsymbol{x}_i) & \phi_2(\boldsymbol{x}_i) & \cdots & \phi_M(\boldsymbol{x}_i)\end{pmatrix}$, and $\mathrm{Tr}\{\cdot\}$ denotes the trace operation.

Using the similar method, the term $\langle y\mathbf{1}^T\boldsymbol{w}\rangle_{\mathcal{D}_t}$ in (15) is given by

$$\begin{aligned}\langle y\mathbf{1}^T\boldsymbol{w}\rangle_{\mathcal{D}_t} &= \boldsymbol{w}_o^T\boldsymbol{\Phi}\boldsymbol{\Gamma}^{-1}[(1-p_1-p_o)\boldsymbol{H}+p_1\boldsymbol{\Phi}]\boldsymbol{w}_o \\ &\quad + \frac{\sigma_\epsilon^2}{N}\mathbf{1}^T\boldsymbol{\Gamma}^{-1}[(1-p_1-p_o)\overline{\boldsymbol{\phi}}+p_1\mathbf{1}]\,,\end{aligned} \tag{21}$$

For the test set error expression (16), $\langle y'\boldsymbol{\phi}^T(\boldsymbol{x}')\boldsymbol{w}\rangle_{\mathcal{D}_f}$ and $\langle y'\mathbf{1}^T\boldsymbol{w}\rangle_{\mathcal{D}_f}$ are given by

$$\langle y'\boldsymbol{\phi}^T(\boldsymbol{x}')\rangle_{\mathcal{D}_f} = (1-p_1-p_o)\boldsymbol{w}_o^T\boldsymbol{H}'\boldsymbol{\Gamma}^{-1}\boldsymbol{H}\boldsymbol{w}_o + p_1\boldsymbol{w}_o^T\boldsymbol{H}'\boldsymbol{\Gamma}^{-1}\boldsymbol{\Phi}\boldsymbol{w}_o \tag{22}$$

$$\langle y'\mathbf{1}^T\boldsymbol{w}\rangle_{\mathcal{D}_f} = (1-p_1-p_o)\boldsymbol{w}_o^T\boldsymbol{\Phi}'\boldsymbol{\Gamma}^{-1}[(1-p_1-p_o)\boldsymbol{H}+p_1\boldsymbol{\Phi}]\boldsymbol{w}_o\,, \tag{23}$$

where $\boldsymbol{\Phi}' = \frac{1}{N}\sum_{i=1}^{N}\begin{pmatrix}\phi_1(\boldsymbol{x}'_{i'}) & \phi_2(\boldsymbol{x}'_{i'}) & \cdots & \phi_M(\boldsymbol{x}'_{i'})\\ \vdots & \vdots & \vdots & \vdots\\ \phi_1(\boldsymbol{x}'_{i'}) & \phi_2(\boldsymbol{x}'_{i'}) & \cdots & \phi_M(\boldsymbol{x}'_{i'})\\ \phi_1(\boldsymbol{x}'_{i'}) & \phi_2(\boldsymbol{x}'_{i'}) & \cdots & \phi_M(\boldsymbol{x}'_{i'})\end{pmatrix}$. Following the common practice, for large N and N', we can assume that $\boldsymbol{H}' \approx \boldsymbol{H}$, $\boldsymbol{G}' \approx \boldsymbol{G}$, $\langle y'^2\rangle_{\mathcal{D}_f} \approx \langle y^2\rangle_{\mathcal{D}_t}$, $\boldsymbol{\Phi}' = \boldsymbol{\Phi}$, $\boldsymbol{D}' = \boldsymbol{D}$, and $\boldsymbol{\Theta}' = \boldsymbol{\Theta}$. The difference between the generalization error of faulty networks and the training error of faulty networks is given by

$$\begin{aligned}\bar{\mathcal{E}}(\mathcal{D}_f)_\beta - \bar{\mathcal{E}}(\mathcal{D}_t)_\beta &= \frac{2(1-p_1-p_o)^2\sigma_\epsilon^2}{N}\mathrm{Tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{H}) + \frac{2p_1(1-p_1-p_o)\sigma_\epsilon^2}{N}\overline{\boldsymbol{\phi}}^T\boldsymbol{\Gamma}^{-1}\mathbf{1} \\ &\quad + \frac{2p_1\sigma_\epsilon^2}{N}\mathbf{1}^T\boldsymbol{\Gamma}^{-1}[(1-p_1-p_o)\overline{\boldsymbol{\phi}}+p_1\mathbf{1}]\end{aligned} \tag{24}$$

From (20)–(24), the MPE formula for estimating the generalization error of faulty networks is given by

$$
\begin{aligned}
\bar{\mathcal{E}}(\mathcal{D}_f)_\beta = (1-p_1-p_o)\mathcal{E}(\mathcal{D}_t) + \frac{p_1+p_o}{N}\sum_{i=1}^{N} y_i^2 + \frac{2(1-p_1-p_o)^2\sigma_\epsilon^2}{N}\mathrm{Tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{H}) \\
+\frac{4p_1(1-p_1-p_o)\sigma_\epsilon^2}{N}\bar{\boldsymbol{\phi}}^T\boldsymbol{\Gamma}^{-1}\mathbf{1} + \frac{2p_1^2\sigma_\epsilon^2}{N}\mathbf{1}^T\boldsymbol{\Gamma}^{-1}\mathbf{1} - \frac{2p_1}{N}\sum_{i=1}^{N} y_i\mathbf{1}^T\boldsymbol{w} \\
+(1-p_1-p_o)(p_1+p_o)\boldsymbol{w}^T(\boldsymbol{G}-\boldsymbol{H})\boldsymbol{w} - 2p_1(1-p_1-p_o)\boldsymbol{w}^T\boldsymbol{D}\boldsymbol{w} \\
+p_1(1-p_1)\boldsymbol{w}^T\boldsymbol{w} + p_1(1-p_1-p_o)\boldsymbol{w}^T\boldsymbol{\Theta}\boldsymbol{w} + p_1^2\boldsymbol{w}^T\underline{\mathbf{1}}\boldsymbol{w} \qquad (25)
\end{aligned}
$$

In (25), most of terms can be obtained form the training set. The only unknown is the variance σ_ϵ^2 of the measurement noise. The variance can be estimated from the Fedorov's method, given by

$$\sigma_\epsilon^2 \approx \frac{1}{N-M}\sum_{i=1}^{N}(y_i - \boldsymbol{\phi}^T(\boldsymbol{x}_i)\boldsymbol{H}^{-1}\frac{1}{N}\sum_{i'=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_{i'})y_{i'})^2. \qquad (26)$$

5 Simulations

We consider two data sets: (i) the sinc function and (ii) a nonlinear autoregressive time series (NAR) [9]. In the sinc function example, the output is given by $y = \mathrm{sinc}(x) + \epsilon$, where ϵ is a zero-mean Gaussian noise with standard deviation $\sigma_\epsilon = 0.15$. The input x is randomly taken from -5 to 5. Both training set and test set contain 200 samples. The

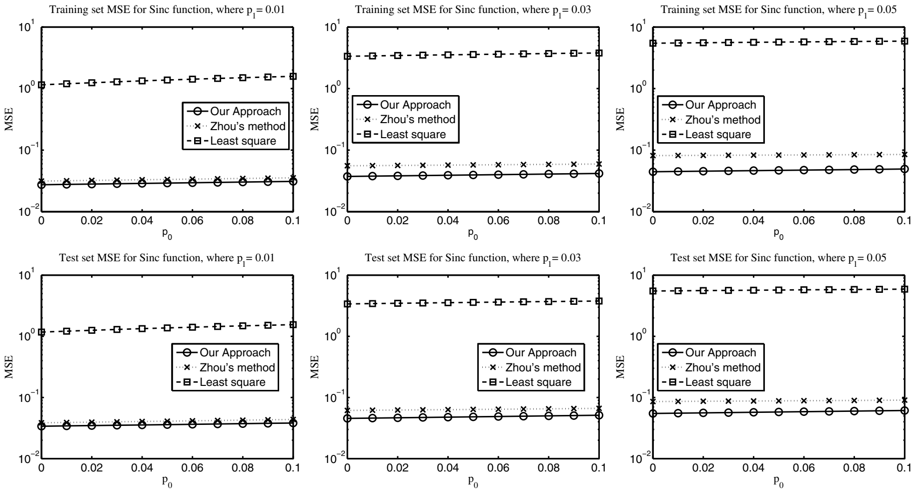


Fig. 1. Training set MSEs and test set MSEs of faulty networks for Sinc function example. Note that the y-axis is in the logarithmic scale.

network model has 37 RBF nodes whose centers are uniformly distributed in the range $[-5, 5]$. The RBF width Δ is equal to to 0.1. For comparison, two other techniques are also considered in the simulation. They are least square and Zhou's method [6]. The least square is a reference which tests the performance of faulty networks when special care is not considered. The average training and test MSEs, under various node fault levels, are shown in Figure 1. The least square method has very poor performance. This result confirms that without special care during the performance of faulty networks could be very poor. The Zhou's method and our approach can improve fault tolerance. Compared with the Zhou's method, our approach has a better performance. The reason is that our approach aims at handling faulty network with the co-existing stuck-at-zero and stuck-at-one faults.

The MPE formula help us not only to estimate the generalization ability of a trained network but also to select some model parameters. For instance, we can use it select the RBF width Δ. Here we illustrate how our MPE results can help us to select an appropriate value of Δ. The NAR example [9] is considered. The output is generated by

$$y(t+1) = \frac{y(t)y(t-1)y(t-2)(y(t-2)-1)x(t-1)+x(t)}{1+y^2(t-1)+y^2(t-2)} + \epsilon(t+1), \quad (27)$$

where $x(t)$ is the input, and the noise term $\epsilon(t)$ is a zero mean Gaussian variable with standard deviation $\sigma_\epsilon^2 = 0.1^2$. We generate 300 samples with $y(0) = y(-1) = y(-2) = 0$. The input sequence $x(t)$'s are random signal generated in the range of $[-0.5{,}0.5]$. The first 150 samples are used for training. The rest of samples is used as test set. The network model has 50 RBF nodes whose centers are randomly selected from the training set. The RBF width Δ is set to 0.6.

Following the conventional approaches in selecting parameters for fault–free networks, we try different values of Δ. Afterwards, we use the MPE formulae to estimate the test error of faulty networks. The results are depicted in Figure 2.

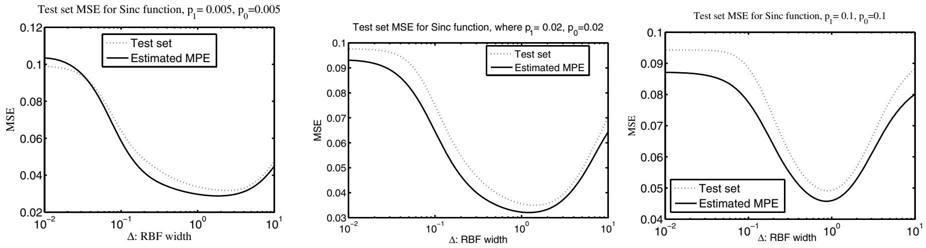


Fig. 2. Use MPE formula to select RBF width for the NAR example

From the figure, although there are small differences between the true test errors [1] and MPE values, our method can locate optimal Δ for minimizing the generalization

[1] When we use the test set method, we need to have a test set and generate a number of faulty networks to measure the performance of faulty networks under different weight noise and weight fault levels.

error of faulty networks. Besides, over a large range of RBF widths, the MPE and test error values are close to the optimal values.

For example, for the NAR example with stuck-at-one fault rate $p_1 = 0.1$ and stuck-at-one fault rate $p_0 = 0.1$, the searched Δ is 0.8710 and the corresponding test set error is 0.04913. When we use the brute force way (test set method) to search Δ, the searched Δ is 0.8912 and the corresponding test set error is 0.04912. When the stuck-at-one fault rate $p_1 = 0.02$ and the stuck-at-one fault rate $p_0 = 0.02$, the searched Δ is 1.2022 and the corresponding test set error is 0.03510. When we use the brute force way (test set method) to search Δ, the searched Δ is 1.4125 and the corresponding test set error is 0.03496. The MPE result confirms the applicability of the MPE formula for the selecting RBF width. For other faulty levels and examples, we obtained similar results (not shown here).

6 Conclusion

This paper addressed the fault tolerance of RBF networks when the stuck-at-zero and stuck-at-one node faults happen at the same time. The performance of faulty networks was investigated. Afterwards, we defined an objective function for minimizing the training error of faulty networks and then developed a learning algorithm. Finally, we derived a MPE formula to predict the generalization performance of the faulty networks trained from our algorithm. With our MPE formula, we can predict the generalization ability of faulty RBF networks without generating a number of faulty networks and without using the test set. Simulation results show that our learning algorithm is better than existing methods tested. Besides, the MPE formula can help us to select an appropriate RBF width value for minimizing the test error of faulty networks.

Acknowledgments. The work was supported by a research grant from City University of Hong Kong (7002760).

References

1. Moody, J.E.: Note on generalization, regularization, and architecture selection in nonlinear learning systems. In: Proc. First IEEE-SP Workshop on Neural Networks for Signal Processing, pp. 1–10 (1991)
2. Phatak, D.S., Koren, I.: Complete and Partial Fault Tolerance of Feedforward Neural Nets. IEEE Trans. Neural Netw. 6, 446–456 (1995)
3. Savich, A.W., Moussa, M., Areibi, S.: The Impact of Arithmetic Representation on Implementing MLP-BP on FPGAs: A Study. IEEE Trans. Neural Netw. 18, 240–252 (2007)
4. Chiu, C.T., Mehrotra, K., Mohan, C.K., Ranka, S.: Modifying Training Algorithms for Improved Fault Tolerance. In: Proceedings of the International Conference on Neural Networks 1994, vol. 4, pp. 333–338 (1994)
5. Leung, C.S., Sum, J.: A Fault-Tolerant Regularizer for RBF Networks. IEEE Trans. Neural Netw. 19, 493–507 (2008)
6. Zhou, Z.H., Chen, Z.H.: Evolving Fault-Tolerant Neural Networks. Neural Comput. Appl. 11, 156–160 (2003)

7. Ho, K., Leung, C.S., Sum, J.: Convergence and Objective Functions of Some Fault/Noise-Injection-Based Online Learning Algorithms for RBF Networks. IEEE Trans. Neural Netw. 21, 938–947 (2010)
8. Bernier, J.L., Ortega, J., Rojas, I., Ros, E., Prieto, A.: Obtaining Fault Tolerant Multilayer Perceptrons Using an Explicit Regularization. Neural Process. Lett. 12, 107–113 (2000)
9. Narendra, K.S., Parthasarathy, K.: Neural Networks and Dynamical Systems. Int. J. Approx. Reason. 6, 109–131 (1992)

Evolutionary Extreme Learning Machine for Ordinal Regression

David Becerra-Alonso, Mariano Carbonero-Ruz,
Francisco José Martínez-Estudillo, and Alfonso Carlos Martínez-Estudillo

Department of Management and Quantitative Methods, Universidad Loyola
Andalucia, AYRNA Research Group
{dbecerra,mariano,fjmestud,acme}@etea.com

Abstract. This paper presents a novel method for generally adapting ordinal classification models. We essentially rely on the assumption that the ordinal structure of the set of class labels is also reflected in the topology of the instance space. Under this assumption, this paper proposes an algorithm in two phases that takes advantage of the ordinal structure of the dataset and tries to translate this ordinal structure in the total ordered real line and then to rank the patterns of the dataset. The first phase makes a projection of the ordinal structure of the feature space. Next, an evolutionary algorithm tunes the first projection working with the misclassified patterns near the border of their right class. The results obtained in seven ordinal datasets are competitive in comparison with state-of-the-art algorithms in ordinal regression, but with much less computational time in datasets with many patterns.

Keywords: ordinal regression, ordinal classification, extreme learning machine, support vector machine, neural networks.

1 Introduction

Ordinal Regression (OR) is a supervised learning problem of predicting categories that have an ordered arrangement. The samples are labeled by a set of ranks with an ordering amongst different categories. In contrast to the nominal classification, there is an ordinal relationship throughout the categories and it is different from regression in that the number of ranks is finite and exact amounts of difference between ranks are not defined. In this way, ordinal classification lies somewhere between classification and regression.

OR problems are important, since they are common in our everyday life where many problems require classification of items into naturally ordered classes. Selecting the best route to work, where to stop, which product to buy, and where to live, are examples of daily ordinal decision-making. In this way, ordinal classification is one of the most important components in many applications.

Compared with general classification problems, much less effort has been devoted to ordinal classification learning. However, in the last decade an increasing number of publications report progress in the artificial learning of ordinal concepts [1,2,3,4,7,9,10,11,15,16,18].

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 217–227, 2012.

Ordinal Classification opens a door that was not readily accessible to us in Nominal Classification: the possibility to somehow project the ordered classes onto a 1D real array. This is made possible by a reasonable a priori assumption [14]: the ordinal outputs to be classified must have a corresponding ordinality in the topology of the instance (or attribute's) space. Ordinal classes are generally expected to overlap in challenging classification problems. Noise and bad quality data are also expected. Yet in OR some degree of coherent gradual transition within the attibutes space is expected. By *coherent,* we mean that there is a correspondence with the a priori class labels provided by the dataset, and the inheretly continuous regressor.

We propose a method that performs OR, including both questionable and reliable patterns, with competitive results. This is basically done by helping the classifier understand the questionable patterns in a way that damps the noise made by them during the classification process.

The Evolutionary Ordinal Regression with Extreme Learning Machine (EOR-ELM) algorithm presented here has two stages. On *Phase 1*, the aim is to project each pattern on a one dimensional real interval in accordance with their classes. The ELM regressor, known to be fast and with good results, is used for this projection of patterns. ELM is used a number of times with different initial weights, in order to keep the one with most accuracy.

The second stage takes on projected values of this better ELM, and begins a sorting process where those patterns incorrectly classified, yet close to their correct class are reallocated in the hopes that this will help that better ELM the next time it performs a regression. An evolutionary criterion is used to reallocate such patterns. The distance to the new location takes place according to random additions or subtractions from the initial values of these patterns, as in random mutation. The new set of mutated and non-mutated projections undergoes ELM once more. Since the input weights for ELM were chosen in *Phase 1*, only the output weights are affected by this reallocation of projected patterns. Robustness is thus ensured, since the regressor remains not significantly altered.

Another important element to remark is how classes are assigned. Previously fixed intervals are not used in this case. Instead the separation between classes is obtained counting the number of patterns on each class. Thus, on this second stage the projections evolve, and so do the boundaries between classes.

Once the second stage is finished, due to a lack of improvement after a number of generations, the boundaries are fixed according to the best evolved classifier (that comes from the best evolved mutated projection).

Although the initial regression is applied to projected patterns within the interval $[0, 1]$, the system is allowed to take on as wide an interval as needed as it evolves in *Phase 2*. This allows for a greater flexibility in the classifier.

The EOR-ELM classifier turns out to be competitive against well-known ordinal classification methods, but most importantly, it provides us with an intuitive means to understand the data we are working with. It is a fast method, allowing for quick convergences in the evolutionary process explained in later sections.

Section 2 presents the nature of the problem, the regressor used (ELM), and how it is used for this particular kind of problem. Section 3 details how the experiments were carried out and their corresponding results. Section 4 ends the document with conclusions.

2 Problem Setup

2.1 The Approach

The center of our proposal is to turn a classification problem into a regression problem so that the class structures are reflected in the regression variable. Let us consider an ordinal classification problem with Q classes that we presume ordered by the class labels, i. e. $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \ldots \prec \mathcal{C}_Q$, and the training set $\mathcal{D} = \left\{(\mathbf{x}_i, y_i) : \mathbf{x}_i \in X \subset \mathbf{R}^k,\ y_i \in Y = \{\mathcal{C}_1, \ldots, \mathcal{C}_Q\}\right\}$ $(i = 1, \ldots. N)$ made by N patterns, where $\mathbf{x}$ is a characteristic random k-vector and y is the class it belongs to.

Our goal is to find a classifier that is capable of assigning, according to the best fit possible, a pattern to its class depending on its characteristics. It should also be designed to include the information related to the ordinality of the classes. Thus, the ordered structure of Y should be used to determine the classifier. This also implies that the order is somehow related to the distribution of patterns in the space of attributes X, and also to the topological distribution of the classes.

We assume the existence of a one-dimensional latent variable $z = \varphi(\mathbf{x})$ that is a function of the characteristics observed and takes on the underlying order mentioned in the previous paragraph.

While ordinality in classification datasets is allowing current researchers to present the problem as a regression where the intervals for each class are chosen one way or another (simple and intuitive), this procedure presents its own drawbacks. First of all, little has been said about how big these intervals must be. We do have the intuition that a given regressor with a given dataset should have an optimal interval for ordinal regression. However it is commonplace to see the interval (0,1) being chosen by default. This is not necessarily always convenient, and it can often make regression harder for a certain method. The second problem has to do with patterns falling close enough to their right interval.

As we can see in Figure 1, patterns may be preserving the ordinality that we are looking for, yet we might be classifying them incorrectly for the sole fact of being (close but) out of their correct interval. In other words, the regressor has done quite a good job ordering the patterns, yet we chose to evaluate it most negatively. In this manner, we are indeed presenting results that look clearly worse than they really are. It is the order of the classes that matters after all, and sticking to interval fitness might be making researchers lose touch with the real challenge (ordering patterns).

We will not use the value for z_j to see if it fits a certain interval. Instead, we will order and rank all the z_j. Then, knowing that class 1 in training had N_1 patterns, we look at the N_1 lowest values returned by the regressor for training. These values we assign to class 1 although they might not belong to it. We do

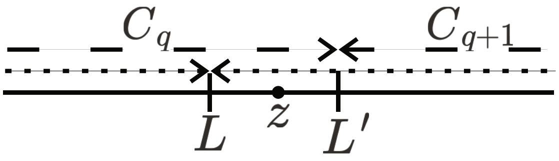


Fig. 1. The pattern projected as z belongs to class C_q. Still, a certain method has placed the boundary L before z, rendering this pattern as incorrectly classified. However, ordinality remains correct when z is immediate to L, since it is right next to its correct class. EOR-ELM proposes a way to define a boundary L' that solves this problem, considering z as correctly classified.

the same for the N_2 values after the lowest N_1 ones, and assign them to class 2, and so on with all the other classes.

2.2 Extreme Learning Machine (ELM)

Huang et al. [12], is the reference for the description of ELM. The regression problem can be formulated as an attempt to find solutions for $\mathbf{w}_i = (w_{i1}, \ldots, w_{in})$ and β_i using the following system of equations:

$$f(\mathbf{x}_j) = t_j, \quad j = 1, 2, \ldots, N \tag{1}$$

where

$$f(\mathbf{x}_j) = \sum_{i=1}^{m} \beta_i g(< \mathbf{w}_i, \mathbf{x}_j > +b_i), \quad j = 1, 2, \ldots, N, \tag{2}$$

where g is the activation function and the symbols $<>$ indicate an ordinary scalar product. This system can also be expressed more concisely as $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$, where $\mathbf{H}$ is the hidden layer's output matrix of the neural network given by:

$$\begin{gathered}\mathbf{H}(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m, b_1, b_2, \ldots, b_m, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_N) = \\ = \begin{pmatrix} g(< \mathbf{w}_1, \mathbf{x}_1 > +b_1) & \ldots & g(< \mathbf{w}_m, \mathbf{x}_1 > +b_m) \\ \vdots & \ddots & \vdots \\ g(< \mathbf{w}_1, \mathbf{x}_N > +b_1) & \ldots & g(< \mathbf{w}_m, \mathbf{x}_N > +b_m) \end{pmatrix}\end{gathered} \tag{3}$$

with

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix}. \tag{4}$$

Each column on matrix $\mathbf{H}$ is made of the values of the corresponding hidden layer node, evaluated for each one of the patterns $\mathbf{x}_i$ in the training set.

The ELM algorithm randomly selects the values for $\mathbf{w}_i = (w_{i1}, \ldots, w_{in})$ and b_i, and then obtains corresponding values for $\beta_1, \ldots, \beta_m$, from the generalized linear model. This is done by calculating the minimum quadratic solution of the linear system, given by:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T} \tag{5}$$

where $\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ is the generalized Moore-Penrose inverse matrix.

In short, the corresponding algorithm for this method is as follows:

ELM Algorithm 1. *Given a training set $D = \{(\mathbf{x}_i, t_i) : \mathbf{x}_i \in \mathbf{R}^n, t_i \in \mathbf{R}, i = 1, 2, \ldots, N\}$, the activation function $g(t)$, and m neurons in the hidden layer:*
Step 1: Assign arbitrary input weights for w and bias b.
Step 2: Calculate the hidden layer output matrix $\mathbf{H}$.
Step 3: Calculate the output weights $\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T}$.

The ELM algorithm has been shown to have a good generalisation capability while it significantly reduces the time needed to train the neural network. For more details on this method the reader can see: [12] and [13].

2.3 The EOR-ELM Algorithm

The EOR-ELM procedure is enumerated as follows:

1. Before the first phase, partition the dataset making sure all classes are almost (at least) equally represented to be used as training and test subsets. Let N be the total number of patterns in the training set. The first N_1 patterns belong to class 1, the next N_2 belong to class 2, and so on until the last N_Q patterns that correspond to class Q. The following will refer to this set.
2. *Phase 1:* A real value is assigned to each one of the patterns. Although these values must be coherent with the order of the classes, there is no a priori information about the order within each class. In order to comply with both interclass labelling and intraclass uncertainty, let us consider the accumulated values $S_0 = 0$, $S_q = \sum_1^q N_i$ and $m_q = \frac{1}{2N}\left(S_{q-1} + S_q\right)$ where $z_i = m_{q(i)}$, for $q\left(i\right) = j$ if $S_{j-1} < i \leqslant S_j$ is assigned. The class that corresponds to each pattern can still be identified according to the arrangement defined in step 1. Thus, the greater z_i correspond to the highest labelled classes (interclass ordinality), and all z_i inside a class are the same (intraclass uncertainty). Indeed, we intend to perform a regression.
3. Invoke a regressor (we use ELM) to be trained according to the new training subset $\{(\mathbf{x}_i, z_i)\ \ i = 1, \ldots, N\}$. ELM will be used M times, each one with different random input weights w. Let φ_w be the regressor and let each pattern be transformed according to $\hat{z}_i = \varphi_w(\mathbf{x}_i)$. Once these outputs are sorted in increasing order, let $r\left(i\right)$ be the rank for $\hat{z}_i$. $C\left(w\right) = \frac{1}{N}\sum \delta\left(q\left(i\right), q\left(r\left(i\right)\right), 0\right)$ where

$$\delta\left(i, j, k\right) = \begin{cases} 1 & |i - j| \leqslant k \\ 0 & |i - j| > k \end{cases}$$

is calculated. Thus C becomes the CCR of the classifier, based on regressor φ_w, that assigns patterns according to the order of its output. Of all the M regressors ran, the one with the highest C is chosen, and called φ.

4. *Phase 2:* An iterated process to improve the regression selected on step 3 is started. Let $k = 0$.
5. Let $z_i^k = \hat{z}_i$. From these values, the interclass boundaries are definde as $L_0^k = z_{(1)}^k$, $L_q^k = \frac{z^k_{(S_q)}+z^k_{(S_{q+1})}}{2}$, $L_Q^k = z_{(N)}^k$ y $m_q^k = \frac{L_q^k+L_{q-1}^k}{2}$, where $z_{(i)}$ is the i-th ranked pattern.
6. The values

$$z_i = z_i^k + \big(\delta\left(q\left(i\right), q\left(r\left(i\right)\right), 1\right) -$$
$$-\delta\left(q\left(i\right), q\left(r\left(i\right)\right), 0\right)\big)\delta\left(i, r\left(i\right), n\right)\left(m^k_{q(i)} - z_i^k\right)v_i,\ v_i \in U\left(0,1\right)$$

are obtained. Here, incorrectly classified borderline (but only n or less units away from the boundary of their class) patterns are randomly reallocated towards the center of the interval of the class they belong to. Incorrectly classified patterns beyond this point are not reallocated, nor are correctly classified patterns. The aim here is to try and reallocate only those patterns close to their correct classification, counting on the inherent continuity of the regressor to not significantly alter those patterns correctly classified (see Figure 2).
7. Accuracy for training is obtained from doing regression to this new z_i class-mutated dataset. The new ELM regressor retains the input weights w obtained in step 3, only changing the output weights β.
8. The accuracies of the original and mutated regressor are compared. The best one is kept, and the other discarded.
9. If the new regressor is chosen, we return to step 5 with $k = k+1$ y $\hat{z}_i = \varphi(\mathbf{x}_i)$. Otherwise we go back to step 6.
10. The stop condition is simply the lack of improvement in accuracy for more than G generations of comparing regressors.
11. The final classifier is

$$\phi\left(\mathbf{x}\right) = q \ \text{ if } L_{q-1} < \varphi\left(\mathbf{x}\right) \leqslant L_q$$

where φ and the boundaries L are the obtained for the regressor at the stop condition, except for $L_1 = -\infty$ y $L_Q = \infty$.
12. The efficiency of the regressor is verified on the test subset, this time only using the boundaries obtained to differentiate classes.

3 Experiments

3.1 Ordinal Classification Datasets and Experimental Design

Up to the author's knowledge, there are no public specific datasets repositories for ordinal classification. The most used dataset repository in the literature is the *ordinal regression benchmark datasets* provided by Chu et. al [4]. However,

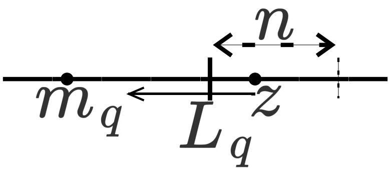


Fig. 2. When and incorrectly classified z ranks among the first n patterns past L_q it can be randomly mutated towards the center of the interval of the class it belongs to (m_q)

Table 1. Datasets used for the experiments

Dataset	Size	#Input	#Classes	Classes Distribution
automobile	205	71	6	(3,22,67,54,32,27)
balance-scale	625	4	3	(288,49,288)
ERA	1000	4	9	(92,142,181,172,158,118,88,31,18)
LEV	1000	4	5	(93,280,403,197,27)
newthyroid	215	5	3	(150,35,30)
SWD	1000	10	4	(32,352,399,217)
tae	151	54	3	(49,50,52)

the benchmark datasets provided by Chu et. al, are not real ordinal classification datasets but regression problems. These datasets are turned from a regression problem into a classification problem by discretizing the target variable into r different bins, with equal frequency or equal width, so each bin is labeled as a different ordinal class.

We have collected a set of real ordinal classification datasets which are publicly available at the UCI repository [8] and at the *mldata.org* datasets repository [17] (see Table 1 for datasets description).

For this method, we perform 10 times a holdout validation and 3 repetitions for each holdout (obtaining a total of $10 \times 3 = 30$ different results). Each holdout is a stratified random division of the data, where approximately 75% of the instances are used for the training set and 25% of them for the test set (maintaining the original distribution of classes for both sets). For the deterministic methods (all of them except EOR-ELM), we perform 30 times a stratified holdout validation using 75% of the instances for the training set and 25% of them for the generalization set, what implies a total of 30 different results. The partitions are the same for all the deterministic methods.

In this way, a total of 30 error measures has been obtained for all the methods compared, which guarantees a proper statistical significance of the results.

3.2 Machine Learning Methods Used for Comparison Purposes

For comparison purposes, different state-of-the-art methods have been included in the experimentation. These methods are the following:

- **Gaussian Processes for Ordinal Regression (GPOR)** by Chu et. al [4], presents a probabilistic kernel approach to ordinal regression based on Gaussian processes where a threshold model that generalizes the *probit* function is used as the likelihood function for ordinal variables.
- **Support Vector Ordinal Regression (SVOR)** by Chu et. al [5][6], proposes two new support vector approaches for ordinal regression. Here, multiple thresholds are optimized in order to define parallel discriminant hyperplanes for the ordinal scales. The first approach with explicit inequality constraints on the thresholds, derive the optimal conditions for the dual problem, and adapt the SMO algorithm for the solution, and we will refer to it as SVOR-EX. In the second approach, the samples in all the categories are allowed to contribute errors for each threshold, therefore there is no need of including the inequality constraints in the problem. This approach is named a SVOR with implicit constraints (SVOR-IM).

Regarding the algorithms' hyper-parameters, the following procedure has been applied. For the Support Vector algorithms, i.e. SVOR-EX and SVOR-IM, the corresponding hyper-parameters (regularization parameter, C, and width of the Gaussian functions, γ), were adjusted using a grid search with a 10-fold cross-validation, considering the following ranges: $C \in \{10^3, 10^1, \ldots, 10^{-3}\}$ and $\gamma \in \{10^3, 10^0, \ldots, 10^{-3}\}$. All the methods were configured to use the Gaussian kernel.

EOR-ELM was run under $M = 300, n = 5, G = 500$. Same order variations of these parameters do not return significant differences in the output.

3.3 Ordinal Classification Evaluation Metrics

Two evaluation metrics have been considered which quantify the accuracy of N predicted ordinal labels for a given dataset $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N\}$, with respect to the true targets $\{y_1, y_2, \ldots, y_N\}$:

1. Mean Zero-one Error (MZE) is simply the fraction of incorrect predictions on individual samples. Accuracy (CCR) measures the correct ones against the entire dataset:

$$MZE = \frac{1}{N} \sum_{i=1}^{N} I\left(\hat{y}_i \neq y_i\right), \tag{6}$$

$$CCR = 1 - MZE, \tag{7}$$

   where $I(\cdot)$ is the zero-one loss function and N is the number of patterns of the dataset.
2. Mean Absolute Error (MAE) is the average deviation of the prediction from the true targets, i.e.:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left|\mathcal{O}(\hat{y}_i) - \mathcal{O}(y_i)\right|, \tag{8}$$

   where $\mathcal{O}(\mathcal{C}_q) = q, 1 \leq q \leq Q$, i.e. $\mathcal{O}(y_i)$ is the order of class label y_i.

These measures are aimed to evaluate two different aspects that can be taken into account when an ordinal regression problem is considered: whether the patterns are generally well classified (CCR) and whether the classifier tends to predict a class as close to the real class as possible (MAE).

Table 2. CCR, MAE and computational time (in seconds), for each dataset and method

Dataset	Method	CCR	MAE	$time/s$
automobile	GPOR	0.4000±0.0633	1.0153±0.1060	26.55
	SVOREX	**0.6788**±0.0608	**0.3788**±0.0821	3.88
	SVORIM	0.6596±0.0573	0.4019±0.0776	**3.77**
	EOR-ELM	0.6677±0.0674	0.3812±0.0789	11.12
balance-scale	GPOR	**0.9726**±0.0095	0.0273±0.0095	978.88
	SVOREX	0.9994±0.0020	**0.0006**±0.0020	111.33
	SVORIM	0.9994±0.0020	**0.0006**±0.0020	38.55
	EOR-ELM	0.9657±0.0197	0.0474±0.0242	**18.89**
ERA	GPOR	0.2812±0.0253	1.2188±0.0732	1356.78
	SVOREX	0.2856±0.0286	1.1804±0.0646	2223.56
	SVORIM	0.2520±0.0184	1.2032±0.0504	3428.78
	EOR-ELM	**0.2965**±0.0186	**1.1535**±0.0525	**36.43**
LEV	GPOR	0.6080±0.0275	0.4172±0.0302	1643.78
	SVOREX	0.6140±0.0293	0.4136±0.0308	1966.33
	SVORIM	0.6124±0.0341	0.4168±0.0340	2137.00
	EOR-ELM	**0.6320**±0.0292	**0.4040**±0.0363	**28.22**
newthyroid	GPOR	0.9463±0.0320	0.0537±0.0320	54.67
	SVOREX	**0.9556**±0.0199	**0.0444**±0.0199	2.89
	SVORIM	0.9538±0.0200	0.0462±0.0200	**1.56**
	EOR-ELM	0.9259±0.0437	0.0926±0.0350	15.07
SWD	GPOR	0.5724±0.0279	0.4464±0.0347	701.56
	SVOREX	0.5660±0.0254	0.4500±0.0301	1026.78
	SVORIM	0.5708±0.0219	0.4448±0.0242	2270.22
	EOR-ELM	**0.5880**±0.0117	**0.4440**±0.0263	**31.40**
tae	GPOR	0.3132±0.0487	0.9736±0.1574	**4.89**
	SVOREX	0.6079±0.0518	0.4421±0.0605	12.44
	SVORIM	0.5974±0.0496	0.4552±0.0582	14.00
	EOR-ELM	**0.6271**±0.0529	**0.4291**±0.0729	13.68

3.4 Results

The model presented in this paper (EOR-ELM), is compared with the ones in Subsection 3.2.The experiments have been carried out by following the experimental design described in Subsections 3.1 and 3.3. Results considering mean and standard deviation in CCR and MAE are showed in Table 2. The best statistical result for each dataset is in bold face. EOR-ELM returns better results in 4 of the 7 datasets. It is particularly competitive in datasets where good classification is hard to obtain. Although EOR-ELM spends most of its computational

Table 3. CCR improvement from *Phase 1* to *Phase 2*

Dataset	*Phase 1*	*Phase 2*
automobile	0.6526	0.6677
balance-scale	0.9586	0.9657
ERA	0.2817	0.2965
LEV	0.6096	0.6320
newthyroid	0.9174	0.9259
SWD	0.5607	0.5880
tae	0.5824	0.6271

time on *Phase 1*, 3 out of 7 datasets were classified faster when EOR-ELM was used.

The relevance of *Phase 2* (steps 4 to 11 in section 2.3) becomes apparent when the increase in accuracy with respect to *Phase 1* is taken into account: a 2.32% increase for automobile, 0.74% for balance-scale, 5.26% for ERA, 3.67% for LEV, 0.92% for newthyroid, 4.87% for SWD, and 7.67% for tae. ELM is a good classifier on its own; EOR-ELM is good at refining the best results provided by ELM (see Table 3). This also indicates the degree of usefulness this method has for each one of the datasets: little use when accuracy is already too close to perfect, and greater on particularly difficult datasets.

4 Conclusions

A novel method for supervised ordinal classification was presented. The approach essentially relies on the assumption that the ordinal structure of the set of class labels is also reflected in the topology of the instance space. A continuous function that projected the instance space onto a real 1D interval (while preserving the ordinality of data) was approximated. This projection of data was carried out in two phases: first, the instance space was projected. Second, the thresholds that define the non-overlapping intervals were determined in an evolutionary process where misclassified patterns near the boundary of their correct class are mutated. The base regressor considered in the process was the Extreme Learning Machine algorithm. This algorithm was able to efficiently project the feature space. The evolutionary tuning process later carried out allowed us to improve the performance of the algorithm. The experimental results obtained show that our algorithm is competitive in comparison with state-of-the-art algorithms in ordinal regression, but with a significant improvement in computational time for datasets with many patterns.

Acknowledgments. This work was supported in part by the Spanish Inter-Ministerial Commission of Science and Technology under Project TIN2011-22794, the European Regional Development fund, and the "Junta de Andaluc ía" (Spain), under Project P2011-TIC-7508.

References

1. Baccianella, S., Esuli, A., Sebastiani, F.: Evaluation Measures for Ordinal Regression. In: Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications (ISDA 2009), pp. 283–287 (2009)
2. Cardoso, J.S., Pinto da Costa, J.F.: Learning to Classify Ordinal Data: The Data Replication Method. Journal of Machine Learning Research 8, 1393–1429 (2007)
3. Cardoso, J.S., Pinto da Costa, J.F., Cardoso, M.J.: Modelling Ordinal Relations with SVMs: an Application to Objective Aesthetic Evaluation of Breast Cancer Conservative Treatment. Neural Networks 18(5-6), 808–817 (2005)
4. Chu, W., Ghahramani, Z.: Gaussian Processes for Ordinal Regression. Journal of Machine Learning Research 6, 1019–1041 (2005)
5. Chu, W., Sathiya Keerthi, S.: New Approaches to Support Vector Ordinal Regression. In: ICML 2005: Proceedings of the 22nd International Conference on Machine Learning, pp. 145–152 (2005)
6. Chu, W., Sathiya Keerthi, S.: Support Vector Ordinal Regression. Neural Computation 19(3), 792–815 (2007)
7. Pinto da Costa, J.F., Alonso, H., Cardoso, J.S.: The Unimodal Model for the Classification of Ordinal Data. Neural Networks 21(1), 78–91 (2008)
8. Frank, A., Asuncion, A.: UCI Machine Learning Repository (2010), http://www.ncbi.nlm.nih.gov
9. Frank, E., Hall, M.: A Simple Approach to Ordinal Classification. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 145–156. Springer, Heidelberg (2001)
10. Hawkes, R.K.: The Multivariate Analysis of Ordinal Measures. The American Journal of Sociology 76(5), 908–926 (1971)
11. Herbrich, R., Graepel, T., Obermayer, K.: Large Margin Rank Boundaries for Ordinal Regression. In: Smola, A.J., Bartlett, P.L., Schölkopf, B., Schuurmans, D. (eds.) Advances in Large Margin Classifiers, pp. 115–132. MIT Press, Cambridge (2000)
12. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. In: IEEE International Conference on Neural Networks - Conference Proceedings, pp. 985–990 (2004)
13. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme Learning Machine: Theory and Applications. Neurocomputing 70(1-3), 489–501 (2006)
14. Hühn, J.C., Hüllermeier, E.: Is an Ordinal Class Structure Useful in Classifier Learning? International Journal of Data Mining, Modelling and Management 1(1), 45–67 (2008)
15. Kramer, S., Widmer, G., Pfahringer, B., de Groeve, M.: Prediction of Ordinal Classes Using Regression Trees. In: Ohsuga, S., Raś, Z.W. (eds.) ISMIS 2000. LNCS (LNAI), vol. 1932, pp. 426–434. Springer, Heidelberg (2000)
16. Li, L., Lin, H.-T.: Ordinal Regression by Extended Binary Classification. Advances in Neural Information Processing Systems 19, 865–872 (2007)
17. Sonnenburg, S.: Machine Learning Data Set Repository (2011), http://mldata.org/
18. Tutz, G., Hennevogl, W.: Random Effects in Ordinal Regression Models. Computational Statistics & Data Analysis 22(5), 537–557 (1996)

Biclustering and Subspace Learning with Regularization for Financial Risk Analysis

Bernardete Ribeiro[1] and Ning Chen[2]

[1] CISUC - Department of Informatics Engineering, University of Coimbra, Portugal
[2] GECAD, Polytechnic Institute of Porto, Portugal
bribeiro@dei.uc.pt, ningchen74@gmail.com

Abstract. Financial models draw on the need to turn critical (economical) information into better decision making models. When it comes to performance enhancement many advanced techniques have been used in bankruptcy detection with good results, yet rarely biclustering has been considered. In this paper, we propose a two-step approach based first on biclustering and second on subspace learning with constant regularization. The rationale behind biclustering is to discover patterns upholding instances and features that are highly correlated. Moreover, we placed great emphasis on building a weight affinity graph matrix and performing smooth subspace learning with regularization. In particular, the geometric topology of biclusters is preserved during learning. Experimental results demonstrate the success of the approach yielding excellent results in a real French data set of healthy and distressed companies.

Keywords: Biclustering, Subspace Learning, Financial Risk Mining, Weight Affinity Graph.

1 Introduction

The interplay between machine learning and knowledge extraction is one example of today's most important developments in computer science. In the midst of the most severe world economic crisis the discovery of patterns in financial data that can uncover firms statuses has a critical impact. The existing methodologies for financial mining apply standard clustering algorithms to group similar patterns of companies. However, these algorithms generally take into account only the global similarities between companies and assign each company to only one cluster, limiting the amount of information that can be extracted. An alternative proposal capable of solving these drawbacks is the biclustering technique. The biclustering algorithm is able to perform clustering of rows and columns simultaneously, allowing for a more comprehensive analysis of financial patterns for the detection of default companies.

There have been a vast number of approaches which apply successful biclustering [1] in a broad range of areas such as text mining [3], biological gene expression [10,4], foreign exchange rate [7] among others. Yet, in financial risk analysis it has never been applied. In this paper we seek to find financial bicluster patterns which simultaneously cluster patterns of attributes (financial

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 228–235, 2012.

ratios) with samples (companies) in a real world data set of French companies. Then we exploit a subspace learning model on the basis of a proper constructed graph weight matrix. Finally, taking advantage of the properties of the projected data a classification model is built with Support Vector Machine (SVM). The experimental results on a real world database of French companies show that the properties of the projected data yield meaningful and appealing visualization and clustering of data. Furthermore, by combining biclustering with subspace learning in a supervised learning manner prior to classification yields desirable results, demonstrating that this approach is very effective.

The paper is organized as follows. Section 2 introduces the biclustering method. The subspace learning approach with smooth regularization (SSSL) will be recalled in section 3. In section 4 the proposed approach is presented. In section 5 the experimental setup is briefly described followed by the results discussion. Section 6 will conclude this paper and suggest further research topics.

2 Biclustering

Traditional clustering aims at finding global patterns by maximizing the similarity within a class and minimizing the similarity between classes. Usually the Euclidean distance function is used. In the case of high dimensional data the curse of dimensionality is likely to occur. A broad-range survey on clustering can be found in [11]. One limitation is that in most of the techniques one object can only belong to one group. This limitation results from the direct selection of attributes. However, one object can belong to the same group from different subsets of attributes. Biclustering alleviates this drawback allowing an object to belong to different groups. By arranging data in such a way that both the samples and attributes are taken into account we come to a technique with proven performance in various kind of problems. Biclustering is an approach that finds local patterns on objects (samples) based on the similarity of attributes (features). The goal of biclustering algorithm is to search highly correlated patterns based on some homogeneity criteria [13]. It was first used in gene expression analysis by Cheng and Church [5]. Algorithms of biclustering include several techniques such as block clustering, coupled two-way clustering, Gibbs sampling, particle swarm optimization among many other [1]. In [8] a survey on the biclustering taxonomy distinguishes the algorithms according to (i) the type of biclusters they find, (ii) the structure of the biclusters, and (iii) the way the biclusters are discovered. The classical approach is to interpret biclustering as a bi-permutation problem so that first rows and then columns are re-ordered to foster clusters in different regions of the original matrix. In another perspective several sub-matrices from the original matrix are created with the goal of maximizing some similarity measures [3].

Suppose we are given a matrix $X \in \mathbb{R}^{m\times n}$, where each element is represented by x_{ij}, $i \in \{1, \cdots, m\}$ is the row index and $j \in \{1, \cdots, n\}$ is the column index. We denote $R = \{1, \cdots, m\}$ and $C = \{1, \cdots, n\}$ the sets of rows and columns, respectively, of matrix X, i.e., X matrix can be described by $X(R, C)$. If we have $I \subseteq R$ and $J \subseteq C$, respectively, as subsets of rows and columns, we denote $X(I, J)$ as the sub-matrix of X containing only the elements x_{ij} with indexes within the sets I and J. By definition, one cluster of rows (or objects) is a sub-matrix of X which contains a certain similarity between their rows for all the attributes. Moreover, a cluster of rows can be then described as $X(I, C)$, where $I = \{i_1, \cdots, i_k\}$ is a subset with $k \leq m$ rows, where $i_r \in R$ and $r \in \{1, \cdots, k\}$ and C is the set of all columns. In a similar way, a cluster of columns (or attributes) is a sub-matrix of X defined by $X(R, J)$, with $J = \{j_1, \cdots, j_s\}$ with $s \leq n$ and such that $j_c \in C$ and $c \in \{1, \cdots, s\}$, and R is the set with all rows where the elements of this sub-matrix show the similarity among them.

From the definitions above a bicluster can be defined as the sub-matrix of X defined by $X(I, J)$ with $I = \{i_1, \cdots, i_k\}$ and $J = \{j_1, \cdots, j_s\}$, where $k \leq m$ and $s \leq n$ whose elements present some sort of similarity to the problem.

The biclustering problem can be formulated as: given the $m \times n$ matrix X find a set of biclusters $B_r = (I_r, J_r)$ with $r = 1, \cdots, t$ where each bicluster B_r satisfies any condition of homogeneity. For the homogeneity aspect, mean square residue score (MSRS) [5] and average value (ACV) [10] are computed as:

$$MSRS = \frac{1}{mn} \sum_{i \in R, j \in C} (x_{ij} - x_{iC} - x_{Rj} + x_{RC})^2 \tag{1}$$

$$x_{iC} = \frac{1}{n} \sum_{j \in C} x_{ij}, \; x_{Rj} = \frac{1}{m} \sum_{i \in R} x_{ij}, \; x_{RC} = \frac{1}{mn} \sum_{i \in R, j \in C} x_{ij} \tag{2}$$

With a homogeneity threshold δ defining the maximum allowable dissimilarity, a valid bicluster can be determined if $MSRS \leq \delta$.

$$ACV = \max \left\{ \frac{\sum_{i,j \in R} |CorR_{ij}| - m}{m^2 - m}, \frac{\sum_{k,l \in C} |CorC_{kl}| - n}{n^2 - n} \right\} \tag{3}$$

where $CorR_{ij}$ and $CorC_{kl}$ are, respectively, the correlation coefficients between rows i and j and columns k and l. A bicluster with high homogeneity in the attributes should have a low MSRS and a high ACV.

3 Spatially Smooth Subspace Learning (SSSL)

Suppose we have m companies described by n financial descriptors (attributes). Let $\{\mathbf{x}_i\}_{i=1}^m \in \mathbb{R}^n$ denote their representation and $X = \{\mathbf{x}_1, \cdots \mathbf{x}_m\}$. Given a graph G with m nodes, each node representing a data point, let W be a symmetric $m \times m$ matrix where W_{ij} is the connection weight between node i and j. Each node of the graph is represented as a low-dimensional vector and the similarities between pairs of data (in the original high-dimensional space) are preserved. The corresponding Laplacian matrix [6] is defined as:

$$L = D - W, \; D_{ii} = \sum_{j \neq i} W_{ij} \quad \forall i \tag{4}$$

where D is a diagonal matrix whose entries are sums of columns (or rows) of the matrix W. Let the low-dimensional embedding of the nodes be $\mathbf{y} = [y_1, y_2, \cdots, y_m]^T$, where the column y_i vector is the embedding for the vertex $\mathbf{x}_i$. Direct graph embedding aims to maintain similarities among vertex pairs by following the graph preserving criterion (6):

$$\mathbf{y}^* = \arg \min_{\mathbf{y}^T D\mathbf{y}=1} \sum_{i \neq j} ||y_i - y_j||^2 W_{ij} \tag{5}$$

$$= \arg \min_{\mathbf{y}^T D\mathbf{y}=1} (\mathbf{y}^T L\mathbf{y}) = \arg \min \frac{\mathbf{y}^T L\mathbf{y}}{\mathbf{y}^T D\mathbf{y}} \tag{6}$$

The similarity preservation property of the graph G follows the idea that if the similarity between samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is high (low), then the distance between $\mathbf{y}_i$ and $\mathbf{y}_j$ should be small (large) to minimize equation (6). Hence, the similarities and differences (among vertex pairs) in the graph are preserved in the embedding [12]. The above optimization problem has the equivalent form below given (4):

$$\mathbf{y}^* = \arg\max\, (y^T W\mathbf{y}) = \arg\max \frac{\mathbf{y}^T W\mathbf{y}}{\mathbf{y}^T D\mathbf{y}} \tag{7}$$

Let $\mathbf{u}$ be the transformation vector and $\mathbf{y}_i = \mathbf{u}^T\mathbf{x}_i$. The optimal $\mathbf{u}^*$ are the eigenvectors corresponding to the maximum eigenvalues of the decomposition problem:

$$XWX^T\mathbf{u} = \lambda XDX^T\mathbf{u} \tag{8}$$

Spatially Smooth Subspace Learning (SSSL) uses the graph structure W and solves the following optimization problem:

$$\mathbf{u}^* = \arg\max \frac{\mathbf{u}^T XWX^T\mathbf{u}}{(1-\alpha)\mathbf{u}^T XDX^T\mathbf{u} + \alpha\mathcal{L}} \tag{9}$$

where $\mathcal{L}$ is the discretized Laplacian regularization function and α is the parameter that controls the smoothness of the approximation.

4 Proposed Approach

As mentioned earlier biclustering consists in simultaneous partitioning of the set of the companies samples (rows data matrix) and of the financial ratios (columns data matrix) into subsets (classes). The biclustering phase results in finding sets of financial ratios similarly expressed in subsets of corporate data. In our model, it works as an extraction of patterns upholding the relevant information for the next step of subspace learning. Following, we build the weight graph matrix W_{ij} where each node of the graph G is a low-dimensional vector found in the biclustering stage, and the weight of edges is calculated upon the local features discovered by the biclusters. Then we build the Lapalacian regularized matrix

L and by direct graph embedding, ensuring the similarity preservation property of the nodes of the graph $G(W, L)$, we find a subspace learning model. Finally, we use the projected data in the SVM classification stage. Figure 1 depicts our approach which can be summarized in the following steps:

1. biclustering of companies and financial ratios
2. construct the weight affinity graph matrix based on discovered biclusters
3. subspace learning with regularization
4. binary classification (healthy, distressed) using SVM
5. track the companies traces in the span of years

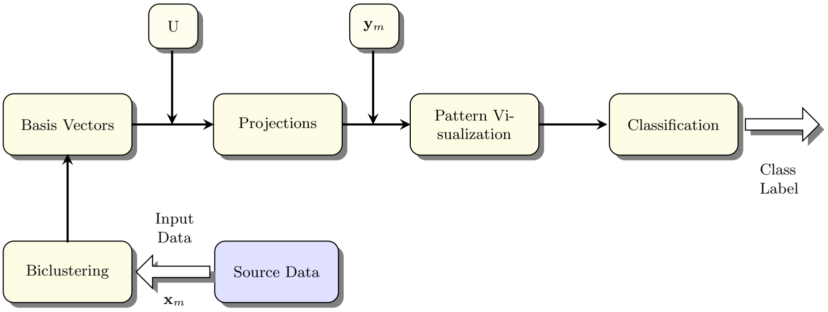


Fig. 1. Biclustering and subspace learning approach

5 Experimental Results

We used Diane database which contains financial statements of French companies. One of the problem goals is to find a model able to predict the class (healthy, bankrupt) in a correct manner. Therefore, bankruptcy prediction is handled as a binary class problem. The initial sample contained about 60000 financial statements from industrial French companies (during the years of 2002 to 2006) with at least 10 employees. From these companies, about 3000 were declared bankrupted in 2007 (or presented a restructuring plan to the court for approval by the creditors). After pre-processing the bankruptcy data set contains 1200 French companies, 600 examples distressed in 2007, and the remainder are healthy. The 30 financial ratios produced by Coface[1] are described in [9]. The affinity graph matrix W is built by assuming that each i-th node corresponds to a given firm $\mathbf{x}_i$. In the p-nearest neighbor graph we then put an edge between nodes i and j if $\mathbf{x}_i$ and $\mathbf{x}_j$ are nearby points, i.e., if $\mathbf{x}_i$ is among the p-nearest neighbors of $\mathbf{x}_j$ and $\mathbf{x}_j$ is among the p-nearest neighbors of $\mathbf{x}_i$. In the experiments below we set $p = 5$ while the heat kernel width has been changed accordingly. Once the affinity graph is constructed, the weight matrix W can be

[1] Coface is one of largest financial groups in France providing Credit Insurance, the Factoring Information & Ratings and Debt Recovery.

specified by means of some weighting schemes such as binary, heat kernel and dot-product [2].

In Figure 2(a) parallel coordinates (PC) plot multi-dimensional data as line segments among parallel axes[2]. Here, along with the axes lines we highlight companies with financial ratios according to the company status. The results were found for a bicluster (599x5) with 599 rows and 5 columns. The threshold was set to 0.9. The results attain $ACV = 0.987$ (98.7%) and $MSRS = 0.0225$ (2.25%) which are good indicators of bicluster quality. The former can be interpreted as accuracy and the latter as error. The financial ratios correspond to columns C16, C24, C28, C29 and C30 in this bicluster, i.e., Cashflow/Turnover (x_{16}), Net Profit Margin (x_{24}), Return on Total Assets (x_{28}), EBIT Margin(x_{29}) and EBITDA Margin (x_{30}). The PC plots show the difference matrix w.r.t. EBIT Margin (C29) where the red line indicates one pattern (sample) in the healthy companies. In Figure 2(b) we plot the results for the biclusters (B1→B19) with varying percentage of companies (% rows) and with 6 financial ratios found by the algorithm. At this point, it is interesting to notice that C23 is the added column, corresponding to the Operating Profit Margin (x_{23}). It shows that this financial indicator plays also an important role in default discrimination. For the same set of experiments reported in Figure 2(b), in Figures 3(a) and (b) we plot, respectively, MSRS with varying noise threshold, and ACV values against MRSV values. As expected it is observed that in Figure 3(a) the error increases by increasing the noise threshold and in Figure 3(b) the higher the accuracy (ACV) the lower the error (MSRS) in the found biclusters (shown points in the graph).

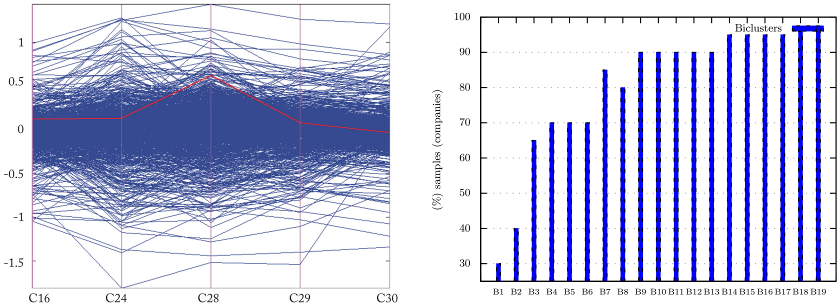


Fig. 2. (a) Parallel Coordinate (PC) Plots. (b) Biclusters with % samples(rows).

A comparative study with and without the biclustering stage was pursued. The results show significant improvement at both levels of performance and visualization. Figure 4 shows the results for the two eigenvectors with highest eigenvalues that result from the projection for (a) subspace learning and (b) biclustering and subspace learning. Analyzing the accuracy results with an SVM, after running the classification step 5 times with 10-fold cross validation, the first method yields 95.6% ± 0.6 and the second method 97.5% ± 0.4. It demonstrates

[2] See http://www.eie.polyu.edu.hk/~nflaw/Biclustering/

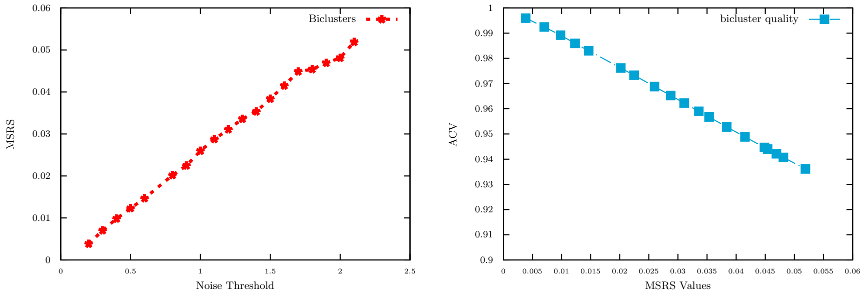


Fig. 3. (a) MSRS versus noise threshold and (b) ACV versus MSRS in found biclusters.

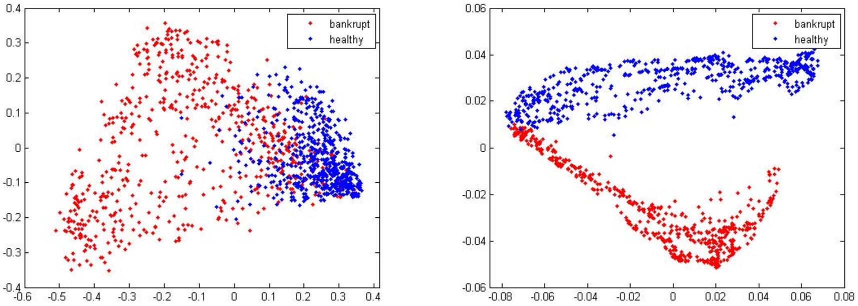


Fig. 4. (a) Subspace with Rank = 6 and (b) Biclustering (6 financial ratios) and Subspace for bankrupt and healthy firms

that the local patterns discovered by biclustering are beneficial to construct a more compact affinity graph, thus enhancing the accuracy of the subsequent classification.

We conducted several experiments varying the heat kernel parameter σ and the regularization constant α for the subspace method by extending our earlier work in [9]. The experimental results show the approach presented herein (by combining biclustering with subspace learning) achieves better performance than the competing approach without biclustering in terms of prediction accuracy. Due to limitation of space they are not included here.

6 Conclusion and Future Work

In this paper, a biclustering model applied on collected financial data from a large set of French companies is developed prior to a subspace learning stage where the geometric properties of (found) biclusters are preserved. Biclustering upholds a local model of clustering while more traditional techniques of clustering attain a global model since they take into account all the attributes. The projected data is clamped into an SVM yielding 97.5% of accuracy while without the biclustering stage there is a drop off to 95.6% of accuracy. Beyond the practical aspect of accuracy improvement in the balanced data set, the visualization of financial data shows evidence from a well-performed learning task. Contrarily to traditional techniques unlikely to find a parsimonious solution, the proposed approach seems quite appropriate in this financial setting. Future work will trace the performance of this approach while spanning over the historic data. It is also valuable to conduct similar studies using other subspace learning methods and classification models in order to further investigate the potential of biclustering in financial risk analysis.

References

1. Busygin, S.: Biclustering in data mining. Computers & Operations Research 35(9), 2964–2987 (2008)
2. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized non-negative matrix factorization for data representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(8), 1548–1560 (2011)
3. de Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J.: Applying Biclustering to Text Mining: An Immune-Inspired Approach. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 83–94. Springer, Heidelberg (2007)
4. Cheng, K., Law, N., Siu, W., Liew, A.: Identification of coherent patterns in gene expression data using an efficient biclustering algorithm and parallel coordinate visualization. BMC 9(210) (2008)
5. Cheng, Y., Church, G.M.: Biclustering of expression data. In: 8th International Conference on Intelligent Systems for Molecular Biology, pp. 93–103 (2000)
6. Chung, F.R.K.: Spectral Graph Theory, vol. 92. American Mathematical Socitey, AMS (1997)
7. Huang, Q.H.: Discovery of time-inconsecutive co-movement patterns of foreign currencies using an evolutionary biclustering method. Applied Mathematics and Computation 218(8), 4353–4364 (2011)
8. Madeira, J., Oliveira, A.L.: Biclustering algorithm for biological data analysis: A survey. In: Workshop on Large-Scale Parallel KDD Systems, pp. 245–260. SIGKDD (2000)
9. Ribeiro, B., Chen, N.: Graph weighted subspace learning models in bankruptcy. In: International Joint Conference on Neural Networks (IJCNN), pp. 2055–2061. IEEE (2011)
10. Teng, L., Chan, L.W.: Biclustering gene expression profiles by alternately sorting with weighted correlated coefficient. In: International Workshop on Machine Learning for Signal Processing, pp. 289–294. IEEE (2006)
11. Xu, R., Wunsch, I.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
12. Yan, S., Liu, J., Tang, X., Huang, T.S.: A parameter-free framework for general supervised subspace learning. IEEE Transactions on Infrmation Forensics and Security 2(1), 69–76 (2007)
13. Zhou, J., Khokhar, A.: ParRescue: Scalable parallel algorithm and implementation for biclustering over large distributed datasets. In: 26th IEEE International Conference on Distributed Computing Systems, pp. 1–8. IEEE Computer Society (2012)

Emotion Understanding in Movie Clips Based on EEG Signal Analysis

Mingu Kwon and Minho Lee*

School of Electrical Engineering and Computer Science, Kyungpook National University,
1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701, South Korea
mgkwon@ee.knu.ac.kr, mholee@knu.ac.kr

Abstract. In this paper, we propose an emotion recognition system for understanding the emotional state of human reflected from a movie clip. In order to analyze the human emotion, we consider the electroencephalogram (EEG) signals which are stimulated while watching movie clips to form the semantic emotional dynamic features. These features are then used to analyze the emotional state of human mind stimulated by emotional scene in movie clips. Changes in alpha and gamma power have been interpreted to indicate differential valence patterns related to the frontal lobes. More active left frontal region indicates a positive reaction, and more active right anterior lobe indicates negative affection. So, the alpha and gamma power in the EEG signals are used to obtain EEG features that recognize the emotional states. In order to extract the emotional feature in a movie clip from EEG signals, both independent component analysis (ICA) which rejects the artifact and Short Time Fourier Transform (STFT) are used. Then, we apply the 3-D fuzzy GIST to effectively describe the emotion related EEG signal. The 3-D fuzzy GIST is based on 3-D tensor data consisting of time dependent energy in a specific power band. The obtained 3-D EEG features are used as inputs to an adaptive neuro-fuzzy inference classifier. We use the mean opinion scores as the teaching signals. Experimental results show that the proposed 3-D EEG features can effectively discriminate the positive emotion from the negative ones.

Keywords: EEG, Positive and Negative Emotion, ICA, STFT, Adaptive Neuro-fuzzy Inference Classifier.

1 Introduction

With the growing interest in human computer interaction, intelligent machines are being developed to satisfy the user's requirements and provide services to improve the quality of the user's life. Nonetheless, a new technology is needed to develop a real human-like intelligent system that is able to provide suitable services to recognize and understand user's emotion states. However, since emotion is a special dynamic form of cognition that is highly complicated, we need to develop a new approach to analyze the human emotions.

* Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 236–243, 2012.

An understanding of the underlying brain dynamics that generate emotional states remains a chaotic area of neuro-science. If one were able to identify emotional feelings and related cognitive processing of associated stimuli from direct cortical neural recordings, it would have broad implications including the potential to be applied for developing the new computer interface system and treating neuropsychiatric disorders with symptoms that include dysfunctional processing of emotional information [1]. If we could obtain knowledge about the specificity of Electroencephalogram (EEG) response patterns in the brain related to primary-processing of emotions, such information may serve as a standard for more accurately identifying typical brain activities. An amount of research into brain correlates of emotion considers them as a generic, more inclusive type of mental function, such as positive versus negative affective groupings. Especially, this research has been further promoted through the use a well-validated set of visual stimuli from the International Affective Pictures System [2]. While there are pictures that represent a number of different emotions, much of the research that uses the pictures for generating responses in EEG has focused on positive vs. negative [3-4]. However, there is a lack of research for analysis of emotional state by dynamic visual stimuli such as movie. Therefore, this paper uses the EEG signal to analyze the state of human emotion while watching a movie.

This paper uses 3-D fuzzy-GIST to obtain the dynamic emotional features, which uses emotional EEG information to extract the emotional factors in EEG and construct the feature space to conduct the human emotion recognition. In order to remove the artifacts from EEG signal, the Independent Component Analysis (ICA) method is used. And to effectively analyze the dynamic human emotions, Short-time Fourier Transform (STFT) is used to obtain a 3-D tensor data consisting of time dependent energy in a specific power band. Based on the human subject feedback feelings evoked by movie clips, the neuro-fuzzy inference system is adapted to learn the 3-D fuzzy GIST features and classify the two different emotions including positive and negative.

The remainder of this paper is organized as follows. The emotion recognition system will be presented in the next section. In section 3, we will present the experimental results and evaluate the performance of the system. Conclusion and discussion are presented in section 4.

2 The Method

2.1 Emotion Recognition System

Fig. 1 describes the overall information flow in the proposed model. The proposed emotion recognition system includes EEG, 3-D fuzzy GIST and an adaptive neuro-fuzzy inference system (ANFIS) in stimuli using movie clips. The system uses the Electroencephalogram (EEG) information obtained while watching the movie clip. ICA which has been widely used for separating non-brain signals such as artifact from the EEG signal [5], is used for preprocessing the EEG signal. Then, STFT is used to acquire both the dynamic characteristic and time-frequency analysis of EEG signal obtained while watching the movie, the 3-D tensor data is used to describe the dynamic emotional characteristics of EEG signal from each short-time block. The tensor data are clustered to make emotional descriptors by fuzzy C-means clustering. Along with EEG information using 3-D fuzzy GIST, the clustering is performed to form primitive knowledge about emotion. Based on the previous knowledge obtained by training procedure, the system learns to understand positive and negative emotions under the control of an ANFIS network. As a result of this procedure, the system can classify the positive and negative emotions.

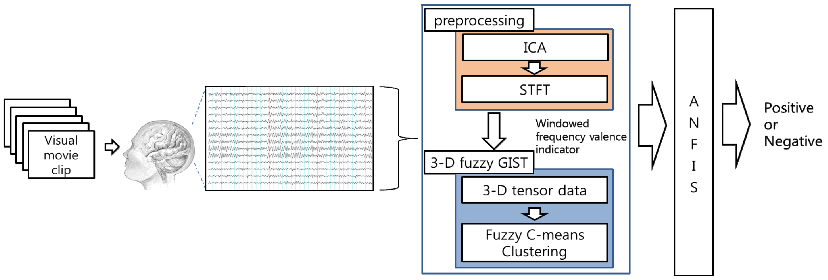


Fig. 1. Graphic outline of the emotion recognition system using the EEG and the 3D fuzzy GIST

2.2 EEG Data Acquisition

The EEG was recorded continuously from 12 electrodes according to the 64 channels EEG system, as shown in Fig. 2 (b), and was digitalized at a rate of 1,250 Hz using a BIOPAC MP150 data acquisition system. EEG changes in alpha power have been interpreted as indicating differential arousal patterns related to the frontal lobes [6]. So the frontal and central areas of the brain were considered for the emotional EEG recording. Fpz was taken as the ground while the linked-earlobe played the role of the reference to reduce the electrocardiography (ECG) artifact. Impedances were kept below 5kΩ. And the data acquisition system was set to trigger at the appearance of the visual stimuli. The participants are stimulated with 6 affective movie clips during the EEG recording. Fig. 2 (a) shows the timing scheme of the paradigm for recording EEG signal using movie clips. First, participants were instructed to maintain gaze on a cross in the center of the video and start when they were ready. After the 5 seconds, an affective movie clip showed up, participants were asked to avoid body movement and view the movie clip until it disappeared 30 seconds later. The EEG recording for each trial ended 5 seconds after the emotional movie clip disappeared. After recording of each trial, we conducted an off-line survey to each subject in order to present emotional states in each movie clip. Then according to opinions, we can assign the movie clips into an emotional flow about each subject.

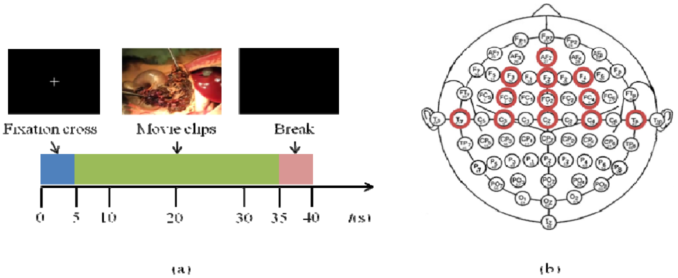


Fig. 2. (a) Timing scheme of the paradigm for recording EEG signal using movie clips; (b) The 64 channels system. The red circles mark the sensor location selected for EEG experiment.

2.3 EEG Signal Processing

The EEG is a powerful non-invasive tool widely used for both medical diagnosis and neurobiological research because it can provide high temporal resolution in milliseconds which directly reflects the dynamics of generating cell assemblies. For any application involving EEG one of the essential and important steps is to remove artifacts due factors such as eye-blinking, muscle noise, and heart signals. One of the commonly used techniques for cleaning of such noises is ICA.

The ICA [5] is a signal processing technique and uses a measure of statistical independence to separate linearly mixed signals that are generated by distinct sources. ICA separates the multi-channel data mixtures into time courses that are maximally independent of one other and in this sense contribute maximally distinct information to the recorded data. Hence, it can be used to de-noise EEG signal when it is mixed with different artifacts such as blinking of eyes. Let us denote by **x** the observed EEG signal whose elements are the linear mixtures $x_1, \cdots, x_n$ and likewise by **s** the random vector of source signals with elements $s_1, \cdots, s_n$. Let us denote by **A** the mixing matrix with elements a_{ij}. All vectors are understood as column vector. Using this vector-matrix notation, the linear mixing model for ICA is written as

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

The ICA finds a demixing matrix **W** to recover the original sources from observed linear mixtures by using information maximization (Infomax algorithm) without any knowledge of the source signals and the mixing matrix [5]. The signal sources separated using ICA are given by

$$\mathbf{u} = \mathbf{W}\,\mathbf{x} \tag{2}$$

The estimated extracted time courses of activation correspond to the independent components. The inverse of estimated weight vector gives relative projection strengths of the extracted components at each of scalp sensors. The strengths of scalp projections provide evidence for physiological origins of the component. The EEGLAB [7] provides a tool for the visualization of component scalp map. Therefore, we obtain the independent components **u** using the EEGLAB developed in MATLAB. After applying ICA, the multi-channel EEG signal is separated in to several independent brain sources. The scalp topographies of the independent components provide information about the location of the sources (e.g., eye activity should project mainly to frontal sites, etc.). “Corrected” EEG signals can then be derived as $\mathbf{x}' = (\mathbf{W})^{-1}\mathbf{u}'$, where **u**' is the matrix of independent components, with rows representing artifactual components set to zero.

After rejecting the artifact by ICA for each trial, we consider Short-Time Fourier Transform (STFT) to understand the positive and negative human emotions from EEG data. STFT, the simplest time-frequency representation, is a two-dimensional representation created by computing the Fourier Transform using a sliding temporal window. By using the STFT [8], we can observe how the frequency of the EEG signals changes with time. As such the details of the resulting short-time Fourier transform are greatly influenced by the choice of windows. In this paper, the Hamming window is used as window function in STFT.

Then, we analyzed EEG signal that indicates the valence state of subject about each blocks by STFT. For emotional valence, psycho-physiological research has shown the importance of the difference in activation between the two cortical hemispheres [9]. Left frontal inactivation is often linked to a negative emotion while right frontal inactivation is a sign of a positive emotion [10]. Changes in alpha power have been interpreted as indicating differential valence patterns related to the frontal lobes [6]. Moreover a significant valence by hemisphere interaction emerged in the gamma band [11]. Therefore, we extracted the power difference between left and right hemispheres in alpha and gamma band to monitor the valence state of test subjects.

2.4 3-D Fuzzy GIST as the Emotional Feature Extraction

We use the 3-D fuzzy GIST based on the fuzzy-GIST [12] to analyze the EEG dynamic emotional features. To effectively analyze the EEG information that have a dynamic characteristic, we need to bundle up the successive time-frequency blocks obtained by STFT and make up cubic type three dimensional tensor which consist of axis such as time, frequency and power. Due to uncertainty in the characteristics of EEG and emotion, we use the fuzzy sets such as Fuzzy C-means clustering (FCM) [13] which have processing of uncertainty. In the proposed system the EEG information is clustered in to four (high positive/low positive/low negative/high negative) respectively using the FCM. By making use of the tensor data, each tensor data is assigned a membership grade using four membership functions in the FCM. Depending on the membership grade, the tensor is assigned to one of the two clusters.

In this model, the obtained emotional dynamic EEG features are used as inputs to an adaptive neuro-fuzzy inference system (ANFIS) [13] to classify the human emotional state. The human brain is also such a powerful system that it can interpret imprecise and incomplete information. Fuzzy set theory provides a systematic approach to process such information linguistically and performs numerical computations using linguistic labels and fuzzy set degrees of membership, which can interpreted as the degrees of truth [13]. The classifier is provided with the mean opinion scores as the teaching signals.

3 Experimental Results

3.1 Environments of Experiments

The experimental visual stimuli employed in the present study consist of videos selected from movies and documentaries. To classify the emotion as either positive or negative, we use 6 videos for the experiments, which are divided into two groups of 3 positive and 3 negative emotional. The test data consists of 2 videos such as 1 positive movie clip and 1 negative one. The test data is randomly selected.

To obtain the target data of the ANFIS network related with the emotional states, the human subjects are asked to answer the following questionnaire after each visual stimulus: 'what was your emotion state after seeing the movie?, Negative or Positive? '

3.2 EEG Experiment

The 11-channel EEG with a sampling rate of 1,250Hz and a total recording time of 40 seconds for each movie clip was used for observing the human brain activity while watching the movie clip. 6 subjects participated in the experiments. During the whole experiment, the subjects were instructed not to move their eyes, or any other part of body but try to stay relaxed and keep the eyes open during the play of movie clip. All the channels were preprocessed by ICA to remove the artifacts and STFT is used to obtain the time-frequency response of EEG to understand the human emotions while watching the movie clip. The width of window function in STFT is 2 seconds and it doesn't have the overlap time among the windows. The power difference between left and right hemispheres in alpha and gamma band were used to indicate the valence state of test subjects. In order to do this, we computed the power spectrum for each channel to measure the power at various frequencies, and we used the average of left and right hemisphere power values at alpha band (8~13Hz) and gamma band(30~60Hz) to get the hemisphere power asymmetry.

3.3 3-D Fuzzy GIST and Experiment Results

The 3-D fuzzy GIST extracts emotional feature by clustering EEG information. First, we bundled up 3 successive time-frequency blocks obtained by STFT and construct the cubic type three dimensional tensors. Second, the FCM is used to partition the emotional EEG information into four clusters. And then, the ANFIS is used to classify the emotional state of human. The subject's feedback is used to generate teaching signals to supervise the learning process of the ANFIS. The ANFIS network was initialized with 4 Gaussian membership functions for each dimension of the input and it searched for the optimal parameters for membership functions and consequent models by parametric tuning, in which the best set of parameters were found by minimizing a sum-squares cost function.

Fig. 3 shows the changes in alpha and gamma power based on the difference in the left and right frontal hemisphere in response to different stimuli such as negative and positive movie clips. It is obtained by computing the average of power values in alpha and gamma power spectrum in each time-frequency response. When the subjects are stimulated by positive emotional movie clips, the alpha and gamma power in frontal lobe is higher than the other case which is stimulated by negative ones. These results indicate the fact that our approach is suitable for understanding the human emotional state while watching the movie clip and can therefore be used by the emotion recognition system. As shown in table 1, the proposed system successfully categorizes the emotional states of human as positive and negative while watching the movie clip. The average performance of the system on the test data is almost 64.75 %. These results show that the proposed system is appropriate to identify the emotional state of human.

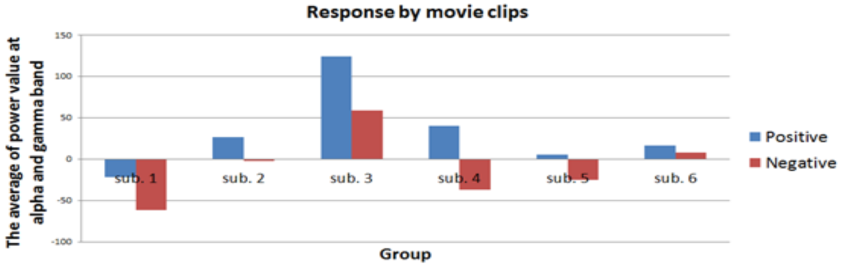


Fig. 3. Changes in alpha and gamma power based on difference between left and right frontal hemisphere in response to different stimuli such as negative and positive movie clips. The blue bars show the result of changes in alpha and gamma power while watching the positive movie clips. The red bars show the result of changes in alpha and gamma power while watching the negative movie clips.

Table 1. The result of the proposed emotion recognition system towards the task of the recognizing the 2 emotional characteristics in the movie clips

	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6
Train (%)	90.38	91.03	92.31	88.46	98.46	99.04
Test (%)	78.85	65.38	59.62	51.92	64.18	68.59

4 Conclusion

A novel developmental scheme for analyzing the human emotions reflected by movie clips was proposed, in which the emotional feature obtained from the subject's brain signal using the 3-D fuzzy GIST. In order to remove the artifacts from the EEG signal, ICA method is adopted. To analyze the emotional state of human from EEG signal, we applied the STFT and the 3-D fuzzy GIST to obtain the 3-D EEG features that are used as inputs to an adaptive neuro-fuzzy inference classifier. In this result, the proposed method is more suitable to recognize the human's emotion in time-variant environment such as real life.

As a future work, we would like to implement an emotion understanding system that can autonomously analyze the human's environment effected by visual and auditory information and immediately can analyze the human's emotional state using EEG signal. So we would like to combine the visual feature using the 3-D fuzzy GIST, EEG feature and auditory feature.

Acknowledgements. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by Ministry of Education, Science and Technology (2012R1A1B6004217) (60%) and the R&D program of the Korea Ministry of Knowledge and Economy (MKE) and the Korea Evaluation Institute of Industrial Technology (KEIT) [10041826, Development of emotional features sensing, diagnostics and distribution s/w platform for measurement of multiple intelligence from young children] (40%).

References

1. Bekkedal, M.Y.V., Rossi III, J., Panksepp, J.: Delineating responses to affective vocalizations by measuring frontal theta event-related synchronization. Neuroscience and Biobehavioral Reviews 35, 1959–1970 (2011)
2. Lang, P., Bradley, M., Cuthbert, B.: International affective picture system (IAPS): Affective ratings of pictures and instruction manual. Technical report A-6, University of Florida (2005)
3. Corr, P.J.: Reinforcement sensitivity theory and personality. Neuroscience and Biobehavioral Reviews 35, 968–978 (2004)
4. Zhang, Q., Lee, M.: A hierarchical positive and negative emotion understanding system based on integrated analysis of visual and brain signals. Neurocomputing 73, 3264–3272 (2010)
5. Jung, T.-P., Makeig, S., Humphries, C., Lee, T.-W., Mckeown, M.J., Iragui, V., Sejnowski, T.J.: Removing electroencephalographic artifacts by blind source separation. Psychophysiology 37(2), 163–178 (2000)
6. Davidson, R.J.: Anterior cerebral asymmetry and the nature of emotion. Brain Cognition 20(1), 125–151 (1992)
7. Delorme, A., Makeig, S.: EEGLAB: An open source toolbox for analysis of single tiral EEG dynamics including independent component analysis. Journal of Neuroscience Methods 134, 9–21 (2004)
8. Hlawatsch, F., Boudreaux-bartels, G.F.: Linear and quadratic time-frequncy signal representations. IEEE Signal Processing Magazine 9(2), 21–67 (1992)
9. Nyemic, C.P.: A theoretical and empirical review of psychophysiological studies of emotion. Journal of Undergraduate Research 1, 15–18 (2002)
10. Davidson, R.J.: Anterior cerebral asymmetry and the nature of emotion. Brain and Cognition 20(1), 125–151 (1992)
11. Müller, M.M., Keil, A., Gruber, T., Elbert, T.: Processing of affective pictures modulates right-hemispheric gamma band EEG activity. Clinical Neurophysiology 110(11), 1913–1920 (1999)
12. Zhang, Q., Lee, M.: Emotion development system by interacting with human EEG and natural scene understanding. Cognitive System Research 14, 37–49 (2012)
13. Jang, J.-S., Sun, C.-T., Mizutani, E.: Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence. Prentice Hall, Inc., Upper Saddle River (1997)

A Distributed Q-Learning Approach for Variable Attention to Multiple Critics

Maryam Tavakol[1], Majid Nili Ahmadabadi[1,2], Maryam Mirian[1], and Masoud Asadpour[1]

[1] Cognitive Robotics Lab, Control and Intelligent Processing Center of Excellence, School of ECE., College of Eng., Univ. of Tehran
[2] School of Cognitive Sciences, Institute for Research in Fundamental Sciences, IPM, Iran
{maryam.tavakol,mnili,mmirian,asadpour}@ut.ac.ir

Abstract. One of the substantial concerns of researchers in machine learning area is designing an artificial agent with an autonomous behaviour in a complex environment. In this paper, we considered a learning problem with multiple critics. The importance of each critic for the agent is different, and attention of agent to them is variable during its life. Inspired from neurological studies, we proposed a distributed learning approach for this problem that is flexible against the variable attention. In this approach, there is a distinct learner for each critic that an algorithm is introduced for aggregating of their knowledge based on combination of model-free and model-based learning methods. We showed that this aggregation method could provide the optimal policy for this problem.

Keywords: Multiple critics, distributed Q-learning, model-based parameters, aggregation method.

1 Introduction

Designing an intelligent system with autonomous decision making ability is one of the principal concerns in machine learning area. In this paper we focus on a learning problem similar to decision making process in human with receiving feedbacks from different sources of reward. A system with multiple users or multiple goals is an example of the application field of this type of problem. We can model this problem as a Reinforcement Learning (RL) process with multiple critics. In RL methods, the agent evaluates its behaviour based on the received reinforcement signals from the environment [1]. In this problem, a set of weights is defined for the importance of critics and the agent attends to each of them according to the corresponding weights. The main problem is that critic's importance can be changed during the agent's life. This is necessary for the learning model to cope with this variable attention without requiring the learning process being done again. While the decision making process of human is flexible to these changes, existing RL methods are not applicable for this purpose.

There is a wealth of research in the domain of human and animal decision making. The neurological studies have revealed the existence of a key RL signal, the

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 244–251, 2012.

temporal difference prediction error, in the brain [2], and some of the RL methods have been applied to these behavioural data [3]. On the other hand, a wide range of neural data suggests that there is more than one system for behavioural choice in the brain. In fact, beside the model-free learning for habitual control, there is a model-based system for goal-directed decision making in the brain [4][5]. So, we can modify RL methods in order to apply them in the problem of variable attention of agent to more than one critic. We consider a distributed approach for resolving the multiple-critic learning problem. A sort of modular approaches for the learning problem have been introduced. Some of them have been designed for a task with multiple goals [6][7][8], while the others have been used for the task decomposition and behaviour coordination of independent subtasks [9][10]. All of these methods assume that the subtasks are completely independent and they do not need an aggregation of the different decisions. In our approach, there is a distinct learner for each reward source and we need a function to aggregate their decisions in each state. In [11], it is shown that in a modular RL, it is impossible to construct an arbitration function that satisfies a few basic and desirable properties for the social choice. Thus, we proposed an algorithm that uses the parameters of the model-based system to address this problem. By combining the model-free and model-based learning methods, we use the advantages of both of them together. Fig. 1 shows how our approach places in learning structure. In the next section we introduce the details of our proposed approach and the correction algorithm, for aggregation of decisions. In section 3, mathematical terms are used to show that how the mentioned algorithm works optimally. Finally we conclude our approach in section 4.

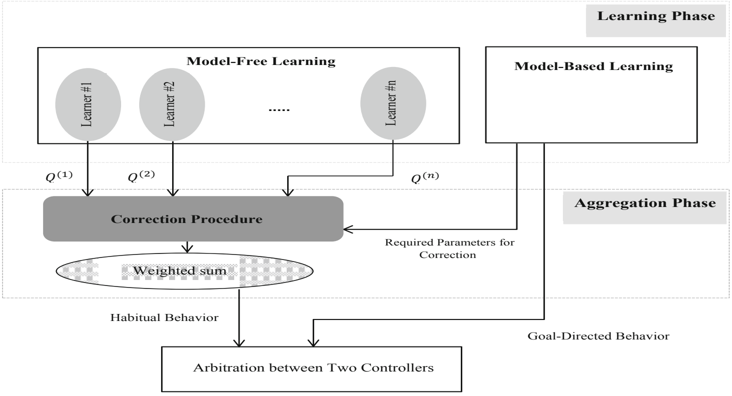


Fig. 1. The position of our proposed approach in the learning process

2 Proposed Approach

In this multiple-critic problem, the received reward from each critic is scaled based on the weight of the corresponding critic. These weights might be changed during the learning life of the agent as the attention of agent to critics changes.

We assume that all the received rewards are summed to form a total reward for learning the action values. However the problem is that if the attention of agent to critics changes, the learned values based on the weighted sum of the rewards will not be valid any more, and the learning has to start from the very beginning. To resolve this problem, a distributed architecture has been introduced that there is a distinct learner for each reward source to learn the Q-value of each state-action pair, independently. In order to apply a learning method to each learner, we preferred an off-policy method over on-policy ones. The Q-values in SARSA, as an on-policy method, rely on behaviour policy of the agent and the current weighted sum of the rewards. The Q-Learning seems a proper method for our purpose. The Q-tables are obtained independent of the behaviour policy and the weights of critics.

2.1 Problem Formulation

In this problem we use the traditional RL framework in an environment with Markov property. In this framework, S is the set of all the possible states and A is all the possible actions. The agent in the state $s \in S$ receives n feedbacks from the n different reward sources, say $r^{(1)}, r^{(2)}, ..., r^{(n)}$ by taking an action $a \in A$. For i^{th} critic, there is $R^{(i)}(s,a)$ as a distribution function for generating the reward sequence. The set of $W = \{w_1, w_2, ..., w_n\}$ indicates the weight of the critics, where $\sum_{i=1}^{n} w_i = 1$. In addition, $P^a_{ss'}$ determines the transition probability between two states. Moreover, $\gamma \in [0,1]$ shows the discount factor for the delayed reward. When the learning is accomplished, the Q-value for the i^{th} learner will be $Q^{(i)}(s,a)$ for each state-action pair. While $Q_{opt}(s,a)$ will be an optimal Q-table obtained from the centralized learning system. By the centralized system we mean a learning process that tries to maximize $E\{\sum_{i=1}^{n} w_i r^{(i)}\}$. Finally, the Q-tables of the learners are combined based on current weight of the critics, and form Q_{dist} as the Q-table of distributed learning system, $Q_{dist}(s,a) = f(w_1, ..., w_n, Q^{(1)}(s,a), ..., Q^{(n)}(s,a))$. We proposed the approach for a problem with one terminal state with different paths. The agent should find the optimal path with the maximum weighted average reward.

2.2 Aggregation Method

The distributed system needs an aggregation function for combining the Q-tables to find the optimal policy. We chose the weighted sum of the Q-tables as Equation 1 Since we used this function in the centralized system for total reward computing and it is simple enough to work with a linear function.

$$Q_{dist}(s,a) = \sum_{i=1}^{n} w_i Q^{(i)}(s,a) \tag{1}$$

This aggregation function does not produce the optimal policy. So, we introduce a correction approach to make (1) to be optimal. The problem of the off-policy character of Q-learning is that the one-step value updates for each learner are computed under the assumption that all future actions will be chosen optimally. While if there is an inconsistent state, the state that the optimal actions of the learners are different, in the path of a state-action to the terminal state, this assumption will not hold. We use the model of the environment to correct their values. As mentioned earlier, there is model-based RL that uses the experiences indirectly to build a model of the environment beside the habitual control. The combination of model-free and model-based decision making trades off between flexibility and computational complexity in one view, and two sources of uncertainty: ignorance and computational noise, in another view [12][13]. As Fig. 1 shows, the model of environment is learned in the model-based system and learned parameters are used in the aggregation phase to modify the Q-tables.

2.3 Correction Algorithm

Consider an agent in a stochastic environment with n learned Q-tables and it is going to make an optimal decision in state s. For each action a, the $Q^{(j)}(s,a)$ may need change for j^{th} learner, if two conditions will be hold: First, there is a non-zero probability in the learned model for reaching to an inconsistent state s' after taking a, say $Pr(s,a,s') > 0$. Second, a_1 will be the optimal aggregated action in the s', but $a_1 \neq \arg\max_{a'} Q^{(j)}(s',a') = a_2$. We assumed that s' has been corrected already. The correction procedure is applied to this Q-value based on the following equation.

$$Q^{(j)}_{new}(s,a) = Q^{(j)}(s,a) + Pr(s,a,s') * \gamma^k(-Q^{(j)}(s',a_2) + Q^{(j)}(s',a_1)), \qquad (2)$$

where k is the number of steps between s and s' in the optimal path. Three sets of parameters are required in the correction procedure that should be determined in the learning phase: The set of all the inconsistent states, IS, the transition probability from each state-action to each $s' \in IS$ and all the intermediate states, $Pr(s,a,s')$, and the number of steps between them, $step(s,a,s')$. Algorithm 1 shows the correction procedure for each state s.

In this algorithm $flag$ specifies the corrected states and θ is a threshold parameter that determines how much it is necessary to do correction procedure. In fact, when the transition probability to an inconsistent state is low, we can ignore this small change in order to have an affordable computation. In addition, it is possible for a state to change from consistent to inconsistent. We ignore this change for the first time but the affected states should be corrected in the next passes. Eventually, these modified Q-tables will be valid until the weights of critics change.

3 Analytical Justification

In this section, we are going to show the optimality of the distributed Q-learning approach in two theorems. The estimation of Q-values in the Q-learning for this problem is obtained based on (3) and (4) for the centralized system and i^{th} learner in the distributed system, respectively [14].

Algorithm 1. Correct(s)

```
Require: IS, Pr, step
  for all $a \in A(s)$ and $s' \in IS$ do
    if $Pr(s,a,s') > \theta$ then
      $Correct(s')$ if it is not corrected yet
      $a' := \arg\max_{a"} Q_{dist}(s', a")$
      for all $j \in \{1,...,n\}$ where $a' \neq a_j : \arg\max_{a"} Q^{(j)}(s', a")$ do
        $Q^{(j)}_{new}(s,a) = Q^{(j)}(s,a) + Pr(s,a,s') * \gamma^{step(s,a,s')}(-Q^{(j)}(s',a_j) + Q^{(j)}(s',a'))$
        if $inconsistent(s) = true$ then
          add $s$ to $IS$
          for all $s" \in S$ and $a" \in A(s")$ where $Pr(s",a",s) > \theta$ do
            $flag(s") = 0$
            $Pr(s",a",s') = Pr(s",a",s') - Pr(s",a",s)$
          end for
        end if
      end for
    end if
  end for
  $flag(s) = true$
```

$$Q_{opt}(s,a) = \sum_{i=1}^{n} w_i R^{(i)}(s,a) + \gamma \sum_{s'} P^{a}_{ss'} \max_{a'} Q_{opt}(s',a') \tag{3}$$

$$Q^{(i)}(s,a) = R^{(i)}(s,a) + \gamma \sum_{s'} P^{a}_{ss'} \max_{a'} Q^{(i)}(s',a') \tag{4}$$

It should be noted that the proofs for the uncertain conditions are straight forward and they are omitted from our proofs because of the space limitation. We illustrate that the following theorems are hold.

Theorem 1. *In the distributed system for each state, s, and action, a, if there is no inconsistent state on their path to the terminal state, (1) and (3) will be equivalent.*

Proof. Consider a consistent state, s, with a as an optimal action of all the learners, say for $i = 1,..,n : a = \arg\max_{a'} Q^{(i)}(s,a)$. Hence, a will be the optimal action in the aggregated form as follows:

$$\forall a' \in A(s) : Q^{(i)}(s,a') < Q^{(i)}(s,a) \quad \Rightarrow \sum_{i=1}^{n} w_i Q^{(i)}(s,a') < \sum_{i=1}^{n} w_i Q^{(i)}(s,a)$$

The mathematical induction is used to show the equivalence of two formulas, backward from the terminal state.

1. $N = 1$: Consider s_1 as a state that exactly takes place before the terminal state and a_1 is the optimal action of the learners. For $i = 1,...,n$:

$$Q^{(i)}(s_1,a_1) = R^{(i)}(s_1,a_1) + \gamma \times 0 \quad \Rightarrow Q_{dist}(s_1,a_1) = \sum_{i=1}^{n} w_i R^{(i)}(s_1,a_1)$$

On the other hand, the Q-value of the centralized system will be as follows where both of them are equal.

$$Q_{opt}(s_1, a_1) = \sum_{i=1}^{n} w_i R^{(i)}(s_1, a_1) + \gamma \times 0$$

2. $N = k$: Assume for the consistent state s_k in k steps before the terminal state, (5) will be hold. Hence, a_k will be the optimal action of two systems.

$$Q_{dist}(s_k, a_k) = Q_{opt}(s_k, a_k) \tag{5}$$

3. $N = k + 1$: Let a_{k+1} takes the agent from the state s_{k+1} to s_k. The Q-value for this state-action pair is obtained based on the following equations.

$$Q^{(i)}(s_{k+1}, a_{k+1}) = R^{(i)}(s_{k+1}, a_{k+1}) + \gamma \times Q^{(i)}(s_k, a_k)$$

In the above equation, $Q^{(i)}(s_k, a_k) = max_{a'} Q^{(i)}(s_k, a')$ will be hold based on the assumption of induction. So, for the distributed and centralized system we have as follows that the result of two systems are equivalent.

$$\Rightarrow Q_{dist}(s_{k+1}, a_{k+1}) = \sum_{i=1}^{n} w_i R^{(i)}(s_{k+1}, a_{k+1}) + \gamma \underbrace{\sum_{i=1}^{n} w_i Q^{(i)}(s_k, a_k)}_{(1) \Rightarrow Q_{dist}(s_k, a_k)}$$

$$Q_{opt}(s_{k+1}, a_{k+1}) = \sum_{i=1}^{n} w_i R^{(i)}(s_{k+1}, a_{k+1}) + \gamma Q_{opt}(s_k, a_k)$$

□

Theorem 2. *If there is at least an inconsistent state between s and the terminal state while the agent takes the action a, the correction procedure will make (1) and (3) equivalent.*

Proof. It is assumed that all the intermediate states corrected already. The correction for the current state-action will be taken by the closest inconsistent state, s_{incns}. Consider, in s_{incns} the following is hold where a is the optimal action after correction:

$$a = \arg\max_{a"} Q^{(i)}(s_{incns}, a") \quad i \neq j \ \ and\ a' = \arg\max_{a"} Q^{(j)}(s_{incns}, a") \quad i = j$$

First, we show the proof for the one step before the inconsistent state. Let s_1 be a consistent state and a_1 is the optimal action of each learner to s_{incns}. So, in the distributed case there is

$$Q^{(i)}(s_1, a_1) = R^{(i)}(s_1, a_1) + \gamma Q^{(i)}(s_{incns}, a) \quad i \neq j,$$

$$Q^{(j)}(s_1, a_1) = R^{(j)}(s_1, a_1) + \gamma Q^{(j)}(s_{incns}, a') \quad i = j,$$

$$\Rightarrow Q_{dist}(s_1, a_1) = \sum_{i=1}^{n} w_i R^{(i)}(s_1, a_1) + \gamma \sum_{i=1, i \neq j}^{n} w_i Q^{(i)}(s_{incns}, a) + \gamma w_j Q^{(j)}(s_{incns}, a'),$$

and for the centralized one,

$$Q_{opt}(s_1, a_1) = \sum_{i=1}^{n} w_i R^{(i)}(s_1, a_1) + \gamma \sum_{i=1}^{n} w_i Q^{(i)}(s_{incns}, a).$$

Hence, the correction based on (2) will make two results equivalent as follows:

$$\begin{aligned} Q_{new}^{(j)}(s_1, a_1) &= Q^{(j)}(s_1, a_1) - \gamma Q^{(j)}(s_{incns}, a') + \gamma Q^{(j)}(s_{incns}, a) \\ &= R^{(j)}(s_1, a_1) + \gamma Q^{(j)}(s_{incns}, a). \end{aligned}$$

In this situation, the optimal action may be changed for j^{th} learner. This change makes the state to be inconsistent.

If we propagate this effect to the s_k and a_k, while the intermediate states are consistent, there will be the following equations for two systems. Clearly, using (2) makes both results to be equivalent.

$$Q_{dist}(s_k, a_k) = \sum_{i=1}^{n} w_i [R^{(i)}(s_k, a_k) + ... + \gamma^{k-1} R^{(i)}(s_1, a_1)]$$

$$+\gamma^k \sum_{i=1, i \neq j}^{n} w_i Q^{(i)}(s_{incns}, a) + \gamma^k w_j Q^{(j)}(s_{incns}, a')$$

$$Q_{opt}(s_k, a_k) = \sum_{i=1}^{n} w_i [R^{(i)}(s_k, a_k) + ... + \gamma^{k-1} R^{(i)}(s_1, a_1)] + \gamma^k \sum_{i=1}^{n} w_i Q^{(i)}(s_{incns}, a)$$

□

Therefore, the correction procedure modifies the Q-values of the state-action pairs one by one if needed and the weighted sum of them gives the optimal policy. But in our algorithm, the agent does not move backward and just the current state-action will be corrected based on the inconsistent states, recursively. So, if a state changes from consistent to inconsistent after the correction of the earlier states, the agent may miss the optimal action for the first round.

4 Discussion

In this paper, we proposed a new learning method for a real problem with multiple critics. We introduced a distributed approach inspired from decision making process in the human brain that considers a distinct Q-learner for each critic. Using the Q-learning as an off-policy method makes the Q-tables to be learned independent of the current importance of critics and any behaviour policy. So, the agent learns just based on the received rewards and its learning result will remain usable when its attention to the critics changes. Then the Q-tables are modified for each set of weights based on the learned model without any relearning. Finally, the weighted sum of them determines the greedy policy of the agent. The combination of model-free and model-based learning is helpful. Because model-free control is inflexible to change, while model-based choices are computationally expensive. Hence, when the learned model is inaccurate the former will be used, while the later will be preferred when there are variations in the environment. We will pursue the effect of inaccurate model in our approach in the future works. In the future works, we will study how changing θ affects the resulting policy. In addition, we will investigate the difference of performance of our approach with the optimal system when each new inconsistent state is ignored for the first time.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT Press (1998)
2. Bayer, H.M., Glimcher, P.W.: Midbrain dopamine neurons encode a quantitative reward prediction error signal. Neuron 47, 129–141 (2005)
3. Niv, Y.: Reinforcement learning in the brain. Journal of Mathematical Psychology 53, 139–154 (2009)
4. Dayan, P., Daw, N.D.: Decision theory, reinforcement learning, and the brain. Cognitive, Affective, & Behavioral Neuroscience 8, 429–453 (2008)
5. Gläscher, J., Daw, N., Dayan, P., O'Doherty, J.P.: States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. Neuron 66, 585–595 (2010)
6. Shelton, C.R.: Balancing multiple sources of reward in reinforcement learning. DTIC Document (2006)
7. Raicevic, P.: Parallel reinforcement learning using multiple reward signals. Neurocomputing 69, 2171–2179 (2006)
8. Sprague, N., Ballard, D.: Multiple-goal reinforcement learning with modular sarsa (0). In: International Joint Conference on Artificial Intelligence, vol. 18, pp. 1445–1447 (2003)
9. Park, K.H., Kim, Y.J., Kim, J.H.: Modular Q-learning based multi-agent cooperation for robot soccer. In: Robotics and Autonomous Systems, vol. 35, pp. 109–122 (2001)
10. Samejima, K., Doya, K., Kawato, M.: Inter-module credit assignment in modular reinforcement learning. Neural Networks 16, 985–994 (2003)
11. Bhat, S., Isbell, C.L., Mateas, M.: On the difficulty of modular reinforcement learning for real-world partial programming. In: Proceedings of the National Conference on Artificial Intelligence, vol. 21, p. 318. AAAI Press (2006)
12. Daw, N.D.: Model-based reinforcement learning as cognitive search: Neurocomputational theories. Evolution, Algorithms and the Brain (2011)
13. Simon, D.A., Daw, N.D.: Environmental statistics and the trade-off between model-based and TD learning in humans. In: Advances in Neural Information Processing Systems, vol. 24, pp. 127–135 (2011)
14. Szepesvári, C.: Algorithms for reinforcement learning. Algorithms for Reinforcement Learning 4, 1–103 (2010)

A Hybrid Approach for Adaptive Car Navigation

Siamak Barzegar[1], Maryam Davoudpour[2], and Alireza Sadeghian[2]

[1] Department of Computer Engineering, Islamic Azad University of Qazvin, Qazvin, Iran
Barzegar@Qiau.ac.ir
[2] Department of Computer Science, Ryerson University Toronto, Canada
mdavoud@scs, asadeghi@ryerson.ca

Abstract. This paper is intended to present a method to find an optimized route with intelligent devices for vehicles. Because the vehicles routing problem is one of the possible applications in which the demands of the driver are not specified, this proposed method will use learning automata and fuzzy logics in dynamic environment in order to learn user behavior to predict future behavior and propose an optimized route for the user. The results show that the proposed route is very close to the user desired one.

Keywords: Fuzzy Logic, Learning Automata, Coloured Petri Net, Car Navigation.

1 Introduction

Since the development of cities and the complication of urban infrastructures, routing has become a more complex and time-consuming task. While individuals were formerly able to navigate a city without the aid of maps in order to get from their respective departure to their destination, they now rely much heavier on navigation systems that can often be found in newer model cars. Finding one's position and alternatively one's destination has therefore benefitted greatly by the use of navigation systems. A car navigation system takes one criterion – such as the shortest time and/or distance – and provides the best possible route to be taken by its user.

Route selection in car guidance systems is a process in which an optimized route is derived based on the departure and destination points, which is illustrated on a map. An optimized route can be defined as the shortest route between two points. Optimized routes usually follow standard algorithms that exist for graph research, although they cannot solely reproduce the knowledge of an experienced driver who is informed of traffic and transportation limitations in every part of a city. There are of course systems that carry out more complex tasks based on several criteria for routing [1].

This paper is intended to present a method for adoptive routing in car navigation systems in order to personalize the optimized route search process based on a driver's interest and destination. The proposed method will aid drivers by incorporating their circumstances on the road in order to propose a path that is deemed more desirable. In addition to that, this paper will also suggest an idea for optimized routing to circumvent a number of problems faced by these systems.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 252–261, 2012.

In the considered methods, the learning ability of learning automata in dynamic environments are used to be able to predict the best route formerly regarding to dynamism of the traffic state and to propose the user. Also we took aid from fuzzy logic to involve driver's interests [2].

We simulate the proposed algorithm with a colored Petri net and compare the achieved information with other results of existing algorithms.

1.1 Algorithm's Problems and Bonds and Present Routing Methods

The majority of research performed in this field presented algorithms in the basis of the Dijkstra algorithm and the A^* algorithm [3, 4]. Size and dimensions of a research graph is very important in saving time of the answer in algorithms and routing graphs have a considerable size (number of knots and their angles are so many) [5, 6]. Therefore the majority of research was intended to optimize the response time.

Considering the mobile car state and due to its aboard safety and also on-time send of essential given orders before passing from existing conjunction in Real Time route, the routing action should be performed immediately [7, 8].

Although some of the proposed algorithms are able to gain the answer to the problem in a Real Time manner, most of routes they compute have a low quality and even if we don't consider this problem, there exist some car guidance systems in routing action which cause the implication of these methods to be impossible. Some of these limitations are as follows [9, 10]:

1.2 Disability of Presented Algorithms for Implementation of Adoptive Research

In the presented algorithms until today, routing action cannot be performed based on different demands of drivers. In the real world, drivers usually consider parameters other than reaching their destination. For example they want to pass by specific points-of-interest such as gas stations, banks, restaurants etc. and/or they want to passsome streets which may have long way but has the sufficient space for the driver and other different matters that implementation and consideration of these parameters is almost impossible in the present algorithms but in real world, if we do not interfere these parameters in our routing, these systems, in fact, won't have the proper efficiency.

1.3 Problems of Interfering Information of Traffic Routes

The algorithms presented for consideration of traffic routes has been considered until now, has more theoretical aspect and the practical implementation is usually impossible and some of those that are implemented do not have the sufficient capability and have some problems. One of their main problems is considering traffic conditions for all links in which traffic information id is sent, in practice, for some of the streets. The mentioned algorithms at alteration time of traffic conditions would forcefully search all links to find the target link. This matter will increase the research action time and this matter is of a high importance in these systems.

2 Learning Automata

Learning automata is an abstract model which randomly selects one action out of its finite set of actions and evaluates it on a random environment, then again evaluates the same action and responds to the automata with a reinforcement signal. Based on this action, and received signal, the automata updates its internal state and selects its next action. Fig.1 illustrates the relationship between an automata and its environment.

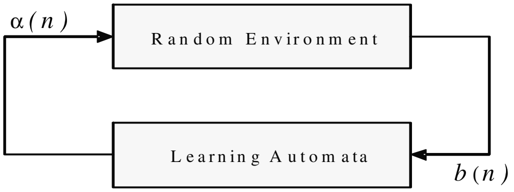


Fig. 1. Relationship between learning automata and its environment

The environment can be defined by $E=\{a,b,c\}$ where $a=\{a_1,a_2,...,a_r\}$ represents a finite input set, $b=\{b_1,b_2,...,b_r\}$ represents the output set, and $c=\{c_1,c_2,...,c_r\}$ is a set of penalty probabilities, and each element c_i of c corresponds to one input of action a_i. An environment in which b can take only binary values 0 or 1 is called P-model environment. Also, by further generalization of the environment it is possible to have finite output sets with more than two elements that take values in the interval [0, 1]. Such an environment is called Q-model environment. Finally, when the output of the environment has continuous random variables, and assumes values in the interval [0, 1], then this environment is known as a S-model environment. Learning automata is classified into stochastic fixed-structure, and stochastic variable-structure. In the following, we only consider variable-structure automata.

A variable-structure automaton is defined by the quadruple $E=\{a,b,p,T\}$ in which $a=\{a_1,a_2,\cdots,a_r\}$ is a set of actions (or outputs of the automaton). The output or action of an automaton is an instant of n denoted by a(n), which is an element of the finite set $a=\{a_1,a_2,\cdots,a_r\}$. $b=\{b_1,b_2,\cdots,b_r\}$ represents the input set or response set, $p=\{p_1,p_2,\cdots,p_r\}$ represents the action probability set, and finally $p(n+1)=T(a(n),b(n),p(n))$ represents the learning algorithm. The following shows, the operation of the automaton based on the action probability set p. The automaton randomly selects an action a_i, and performs it on the environment. After receiving the environment's reinforcement signal, the automaton updates its action probability set based on (1) for favorable responses, and (2) for unfavorable ones.

$$\begin{aligned} p_i(n+1) &= p_i(n) + a.(1-p_i(n)) \\ p_j(n+1) &= p_j(n) - a.p_j(n) \qquad \forall j \;\; j \neq i \end{aligned} \tag{1}$$

$$\begin{aligned} p_i(n+1) &= (1-b).p_i(n) \\ p_j(n+1) &= \frac{b}{r-1} + (1-b)p_j(n) \qquad \forall j \;\; j \neq i \end{aligned} \tag{2}$$

Where a and b are reward and penalty parameters, respectively. If a=b, the automaton is called LRP. If b=0 the automaton is called LRI and if 0<b<a<1, the automaton is called $L_{R\varepsilon P}$. More information about learning automata can be found in [11].

3 Coloured Petri Net

Coloured Petri Nets were introduced by Kurt Jensenin 1987 as a developed model of Petri Nets [12]. Coloured Petri Nets are appropriate tools for mathematical and graphical modeling. Coloured Petri Nets have numerous applications, and lots of research has taken place with respect to modeling, describing and analyzing systems, which have synchronized, asynchronized, distributed, parallel, non-deterministic or random natures. In fact, Petri Nets are models which could represent the performance and state of the system at the same time. There has been enormous research done in the following areas, (i) controlling and learning systems using coloured Petri Nets, (ii) optimizing Petri Net structures using genetic programming and (iii) learning and reasoning the ambiguous problems using fuzzy coloured Petri Nets. However, there are no records for adapting coloured Petri Nets and using learning automata in Petri Net.

A formal definition of a FCPN is in [13]:

The structure of Fuzzy Coloured Petri Nets depends on the fuzzy production rules. The composite fuzzy production rules could be distinguished into following three rule-types respectively [12], [13].

Type1: Simple fuzzy production rule [14,15]:
IF d_j THEN d_k (CF=u)

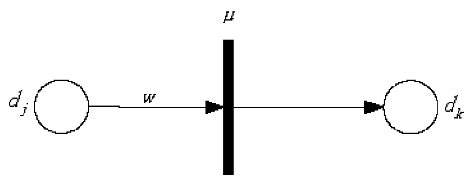


Fig. 2. The FCPN denotation of fuzzy Coloured rule of Type 1

Type2: Compound joined fuzzy production rule:
If d1 AND d2 AND ... AND d_n THEN d_k(CF = u)

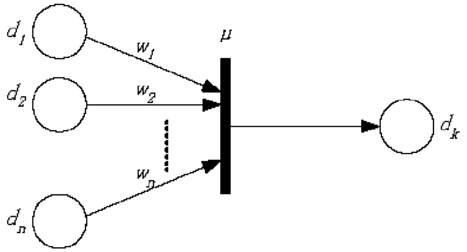


Fig. 3. The FCPN denotation of fuzzy Coloured rule of Type 2

Type3: Compound disjoined fuzzy production rule:
If d1 OR d2 OR ... OR d_n THEN d_k(CF = u)

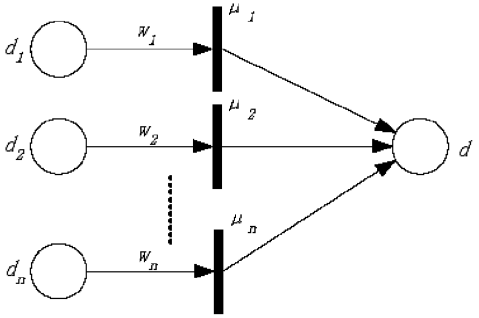


Fig. 4. The FCPN denotation of fuzzy Coloured rule of Type 3

4 The Proposed Algorithm

In this section, a combined algorithm is proposed for routing. Here, Fuzzy logic and learner automats are used for optimized routing. In the proposed algorithms, the learner automats are used for arrangement of membership functions of input and output parameters. A fuzzy variable and a membership function is considered for each of vocal amounts. Each membership function is equipped with a learner automat with a variable structure whose responsibility is to arrange the fuzzy function parameters. Fuzzy membership functions have the types of triangle and trapeze. The initial and final points of membership functions are constant and pre-defined. The responsibility of these learner automats is to arrange membership function center in such a way that it selects the most favorite route regarding to driver's interests. In the proposed algorithm, 10 actions are defined for each of these automats. The selection possibilities of any action of these learner automats in moment of learner process start have the possibility of one tenth.

The proposed algorithm implements in beginning of each conjunction and presents the optimized proposed route based on the beneath specification.

Regarding the presented problems in car routing specially in field of limitation n sending the traffic situation in all conjunctions, we solve this problem via other cars. For example suppose that a car series had stop being the red light or are passing it (the beneath figure), out target car is A, in this state car A reacts with other cars present in conjunction which can pass from a part of routes where the target car pass. If the reacted car has passed from the common route, we gave a coefficient between zero and one to the traffic state dependent on the time it has passed from that route.

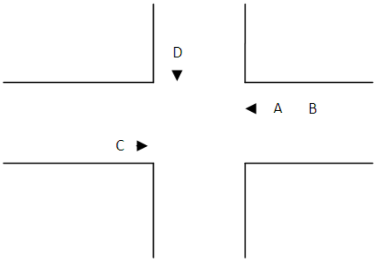


Fig. 5. Four- phase transition

- In each conjunction, the state of traffic light of next conjunctions, which may exist on one of the proposed routes, are examined.
- Considering the demanded route of drivers based on their prior and received information. (For example using the camera set on the forward glass, the driver condition can be processed and it can effect on selection of the proposed route. For example if the driver was angry, the empty route has the highest priority. Also we can ask the idea of the driver at the beginning so the driver can make the routing system informed of his interests and needs. As an example, if he needs gas station, he announces and the algorithm proposes the best route in which exists gas station based on the amount of present gas or if it id lunch or dinner time, it proposes the route in which exists a better one in favor of driver based on present information in internet.)
- Considering the distance amount till the next conjunction
- Considering the number of main conjunctions till destination
- Considering route type from viewpoint of passage amount from main street, by-street and freeways.
- Considering the distance amount in basis of kilometer till destination

To gain the optimized route, all the explained specifications should first turn into input parameters to fuzzy variable and system output is the measurement of favorability in routes. Input and output parameters have their specific vocal parameters.

Table 1. features and their linguistic descriptions

Variables	linguistic descriptions			
Distance to Destination (D_{DQ})	Very Few	Few	Long	Too Long
Quantity of Crosses (Q_C)	Few	Moderate	Many	Too Many
Highway passage (P_W)	Few	Moderate	Many	Too Many
Traffic status according to traffic signal information (T_{IF})	Few	Moderate	Many	
Traffic status based on comparative vehicle information Best regards (T_{CCI})	Few	Moderate	Many	

Fuzzy membership functions a re either triangular or trapezoidal in shape shown in Fig. 6.

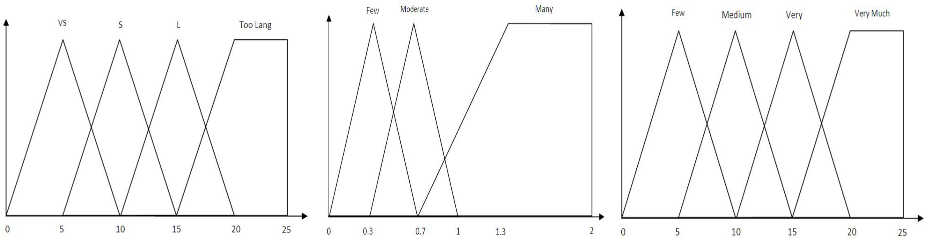


Fig. 6. .Membership functions of input variables

Follows as are parameters output:

Awful	Very Bad	Bad	Good	Very Good	Perfect

Some of the fuzzy production rules are as follows:
IF D is Very and QC is Too Many Then Route is Awful
IF P is Too Many Then Route is Perfect
IF TIF is Moderate and P is Many Then Route is Good
IF TCCI is Moderate and QC is Many Then Route is Bad
IF QC is Moderate and TIF is Few Then Route is Very Good
IF D is Very and QC is Too Many And TCCI is Many Then Route is Very Bad

4.1 Details of the Proposed Algorithm

1. Repeat 10,000 times
2. Each of learner parameters select one of their operation regarding to their probability vector which eventually creates 3 to 4 membership function for each of input parameters.

3. Membership degree of any of the input parameters are computed by attention to the gained information via gained traffic state from conjunctions and the adapted cars and also the gained information from route state from view of number conjunctions and passage from freeway and the created membership functions in previous pace.
4. Considering the amount of gained membership for input parameters and the activated fuzzy rules, the favorability amount of all existing routes for reaching to destination is determined and the route having the most favorability (most of the output amount from membership functions) is selected as the proposed route.
5. Then penalty and reward is accounted in terms of criteria and driver's interests:
   (a) If the proposed route, has considered the driver's interest higher than 80%, a reward is accrued to the proposed route, otherwise penalty is accrued to the proposed route based on the percentage amount of ignorance to driver's interests.

Finally, a route having the most favorability is announced as the optimized route to the driver based on the amount of final favorability gained in 10.000 times of repetition in algorithm.

It is noticeable that the algorithm of finding the ways to destination is not examined in the proposed algorithm and we supposed that the entire existing route from departure to destination points were present.

5 Simulation

In this simulation, the proposed method has been reviewed on several different paths leading to a specific destination. In this method each token represents learning automata for each membership function of a S-$L_{R\varepsilon P}$ learning automata have been used with different parameters, the best result is given by a = 0.1 , b = 0.05 values . Each learning automata has 10 actions with initial probability of 0.1. In this simulation actions are selected based on probability vectors of each learning automata .If in the n^{th} iteration, action α_I is selected where environment response is $\beta_i(n)$, the probability vectors of automata are as follows:

If the queue gets worse environment gives penalty: $\beta_i(n)$ =1 and if it gets better it gives rewards.

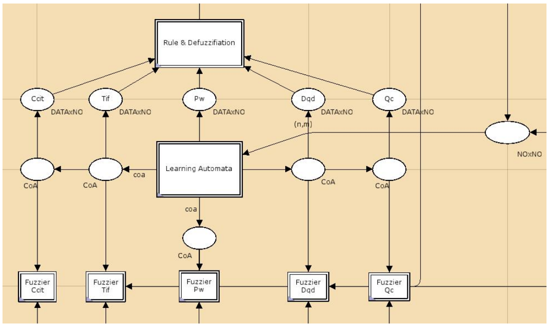


Fig. 7. The Simulation

Table 2. The Result

Penalty rate	Reward rate	Iteration	Adaptively
0.05	0.1	20	94%
0.01	0.1	20	90%
0.005	0.1	20	82%

6 Conclusion

In this paper, an adaptive fuzzy coloured Petri Net has been presented based on learning automata and the applied proposed adaptive model in adaptive routing were studied and evaluated. The proposed algorithm is based on combination of fuzzy logic and learning automata, which has used learning automata to adjust membership function in fuzzy system.

Adaptive model is tried to predict the next optimum mode and update the current system status based on retrieved data from previous events and system responses. In order to evaluate, prior to any test, driver's desired rout is determined, the results of the proposed algorithm shows that the selected rout by the algorithm is very close to driver's desired rout.

Reference

1. Afshordi, N., Meybodi, M.R.: Tuning Fuzzy Membership Functions in Learning Driver Preferences Using Learning Automata. In: Proceedings of International Conference on Intelligence and Advance Systems (ICIAS 2007), Kuala Lumpor, Malaysia, pp. 25–28 (2007)
2. Barzegar, S., Davoudpour, M., Meybodi, M.R., Sadeghian, A., Tirandazian, M.: Traffic Signal Control with Adaptive Fuzzy Colored Petri Net based on Learning Automata. In: Proceedings of 29th North America Fuzzy Information Processing Society Annual Conference (NAFIPS 2010), Ryerson University, Toronto, Canada, pp. 12–14 (2010)
3. Korf, R.E.: Real-Time Heuristic Search. Artificial Intelligence 42(2-3), 189–211 (1990)
4. Hiraishi, H., Ohwada, H., Mizoguchi, F.: Intercommunicating Car Navigation System with Dynamic Route Finding. In: Proc.of IEEE/IEEJ/JSAI Int. Conference on Intelligent Transportation Systems, pp. 284–289 (1999)
5. Chabini, I., Lan, S.: Adaptation of the A* Algorithm for the Computation of Fastest Paths in Deterministic Discrete-Time Dynamic Networks. IEEE Trans. Intelligent Transportation Systems 3, 60–74 (2002)
6. Chan, E.P.F., Zhang, N.: Finding Shortest Paths in Large Network Systems. In: Proc. of the 9th Int. Conference on Advances in Geographic Information Systems. ACM Press, New York (2001)
7. Flinsenberg, I.: Graph Partitioning for Route Planning in Car navigation Systems. In: Proc. of the 11th IAIN World Congress, Smart Navigation Systems and Services, Berlin, Germany (2003)

8. Kim, K., Seungwon, Y., Sang, K.C.: A Partitioning Scheme for Hierarchical Path Finding Robust to Link Cost Update. In: Proc. of the 5th World Congress on Intelligent Transportation Systems, Seoul, Korea (1998)
9. Hashemzadeh, M., Meybodi, M.R.: A Fast and Efficient Route Finding Method for Car Navigation Systems with Neural Networks. In: The Tenth International IEEE Enterprise Distributed Object Computing Conference, EDOC (2006)
10. Hahne, F., Nowak, C., Ambrosi, K.: Acceleration of the A*-Algorithm for the Shortest Path Problem in Digital Road Maps. Operations Research Proceedings, 455–460 (2007)
11. Barzegar, S., Davoudpour, M., Meybodi, M.R., Sadeghian, A., Tirandazian, M.: Formalized Learning Automata with Adaptive Fuzzy Colored Petri Net and Application Specific to Managing Traffic Signals. ScientiaIranica, Transaction D 18(3), 554–565 (2011)
12. Jensen, K., Kristensen, L.M., Wells, L.: Coloured Petri nets and CPN tools for modelling and validation of concurrent systems. International Journal of Software Tools Technology Transfer, 213–254 (2007)
13. Yeung, D.S., Liu, J.N.K., Shiu, J.N.K., Fung, G.S.K.: Fuzzy coloured Petri nets in modelling flexible manufacturing systems. In: ITESM, pp. 100–107 (1996)
14. Ouchi, Y., Tazaki, E.: Learning and reasoning method using fuzzy coloured Petri nets under uncertainty. In: Proc. of the IEEE International Conf. on Computational Cybernetics and Simulation, vol. 4, pp. 3867–3871 (1997)
15. Zhou, C., Jiang, Z.: Fault diagnosis of TV transmitters based on fuzzy Petri nets. In: Proc. of the IMACS Multiconference on Computational Engineering in Systems Applications (CESA), pp. 2003–2009 (2006)

Low Complexity Classification System for Glove-Based Arabic Sign Language Recognition

Khaled Assaleh[1], Tamer Shanableh[2,*], and Mohammed Zourob[1]

[1] American University of Sharjah, Department of Electrical Engineering, Sharjah, UAE
[2] American University of Sharjah, Department of Computer Science and Engineering, Sharjah, UAE
{kassaleh,b00025320,tshanableh}@aus.edu

Abstract. This paper presents a low complexity classification approach for sign language recognition using sensor-based gloves. Each glove includes 5 bend sensors and a 3D accelerometer. The classification approach is based on a novel feature extraction method based on accumulated differences (ADs). The ADs approach projects the dynamics of the glove sensor readings into one feature vector. This vector is normally of high dimensionality as it is meant to capture the dynamics of a sign language gesture. As such, dimensionality reduction using stepwise regression is applied to feature vectors before classification. Thereafter, a simple minimum distance classifier is employed. The proposed system is applied to a dataset Arabic sign language gestures and it yielded a recognition rates 92.5% and 95.1% for user dependent and user independent models respectively. Moreover, the computational complexity of the proposed method is O(N) as compared to the classical approach of Dynamic Time Warping (DTW) which is of O(N^2) complexity.

Keywords: Arabic sign language recognition, Sensor gloves, Dynamic time warping, Accumulated differences.

1 Introduction

Human–computer interaction is a multidisciplinary research area with diverse applications in robotics control, psychological behavior studies, emotion analysis, sign language recognition, assistive e-learning technologies and virtual environments navigation. Human–computer interaction is constantly defining new modalities of communication, and new ways of interacting with machines. One important application of human-computer interaction is sign language recognition that allows for the communication between hearing and deaf parties.

There are two major approaches to gesture recognition; vision-based and sensor-based methods. There are a number of factors affecting the accuracy of the vision-based approach such as lighting, position of the signer relative to the camera and the background movements. Moreover, the dimensionality of the extracted features in a

[*] Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 262–268, 2012.

vision-based system is typically prohibitive, thus affecting the overall complexity of the systems [1].

In general, gesture recognition systems are not 100% accurate and require high computational complexity. In the literature, there have been several attempts towards developing gesture recognition systems using sensor-based methods. Most of which used a small number of features and there have been very few attempts to use sensor-based methods for Arabic sign language recognition, which is the focus of this paper.

In the work reported in [2], the authors presented an online construction algorithm for constructing fuzzy basis function classifiers that are capable of distinguishing different kinds of human daily activities. The activity recognition is based on the acceleration data gathered from a wireless tri-axial accelerometer module raised on users' dominant wrists. To test their approach, eight common domestic activities were used, standing, sitting, walking, running, vacuuming, scrubbing, brushing teeth, and working at a computer. Acceleration readings were gathered from 7 subjects in the age range of 20-25 years old. The recognition accuracy of eight daily activities was found to be 93%.

In [3], the authors presented uWave which is an efficient recognition algorithm, focusing on gestures whilst disregarding fingers movement. This is achieved through using only one tri-axial accelerometer. uWave only requires a single training sample for each gesture. uWave matches the accelerometer readings for an unknown gesture with those for a vocabulary of known gestures, or templates, based on dynamic time warping (DTW). The core of uWave includes dynamic time warping (DTW) to measure similarities between two time series of accelerometer readings; quantization for reducing computation load and suppressing noise and non-intrinsic variations in gesture performance; and template adaptation for coping with gesture variation over the time. The results showed a high accuracy of 98.4%. Two main applications were proposed in their paper, gesture-based 3D mobile user interface and gesture-based user authentication.

The work in [4] introduces an approach for constructing neural classifiers that are able to classify human activities using a tri-axial accelerometer. The acceleration data was collected using a wireless sensing tri-axial accelerometer module mounted on the dominant wrist. In general, it is hard to recognize activities using only one accelerometer, but this difficulty is tackled by using an effective design procedure that consists of data pre-processing, feature extraction, efficient feature subset selection, and neural classifier construction. The new idea in this paper is to divide the dynamic activities and the static activities at the first early stages. This idea is called divide-and-conquer strategy. This approach recognizes these two different types of activities separately. They used multilayer neural networks in the process of activity recognition.

The work in [5] obtained a 3D data about the position of finger tips and used it to train a neural network to predict the fingers posture. In other words, neural networks are proposed to learn the inverse kinematics mapping between the fingertip 3D position and the corresponding joint angles. Using a data glove, training data is obtained. The maximal mean error between fingertip measured position and fingertip position obtained from simulated joint angles and forward kinematics is 0.99±0.76

mm for the training set and 1.49±1.62 mm for the test set. Also, the maximal RMS error of joint angles prediction is 2.85° and 5.10° for the training and test sets respectively, while the maximal mean joint angles prediction error is −0.11±4.34° and −2.52±6.71° for the training and test sets, respectively. It is clear that relatively high gesture recognition rates are applicable using sensor-based methods. However, there still exist problems such as the computation complexity and the ease of use. Moreover, these advances still fall short in helping Arabic sign language. In order to translate the sign language, hand gestures need to be accurately captured in order to ultimately process them.

In this paper, we introduce a sensor-based system for recognizing Arabic sign language. The solution is based on eliminating the time dependency of the data and projecting the captured gesture sequence into one or two static vectors. We have considered both user independent and user dependent modes. This paper is organized as follows. Section 2 describes the database used in this paper. In section 3, we describe the proposed methodology. The experimental results are presented and discussed in section 4. Finally the paper is concluded in section 5.

2 Database Description

We have used the DG5-VHand data glove which is a complete and innovative sensor based system [6]. It has five embedded bend sensors that facilitate accurate measurement of finger movements. It also contains an embedded 3 axes accelerometer which allows sensing both the hand movements and the hand orientation (roll and pitch). The glove communicates with external devices wirelessly via Bluetooth. It has been developed for wireless and autonomous operations and it can be powered with a battery, guaranteeing a long operative period.

A database comprising 10 isolated Arabic sign language gestures performed by 10 different users, repeated 10 times with users wearing the gloves. Hence, a total of 1000 gestures are collected. The sensor readings were acquired at a rate of 30 readings per second. No restrictions on gesture performing speed are imposed.

3 Proposed Methodology

3.1 Feature Extraction

The data acquisition system yields a sequence of 16th dimensional vectors. Each vector comprises 5 bend sensor readings and 3-3D acceleration readings per hand. As we mentioned above, the sampling rate is 30 readings per second per sensor. A nominal length of a gesture is about 5 seconds (i.e. 150 vectors). For example, a sequence of *N* feature vectors representing a certain gesture *i* is represented by the matrix $\mathbf{X}_i$, such as

$$\mathbf{X}_i = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_N]^{\mathrm{T}} \tag{1}$$

The process of extracting the representing feature vector from the above matrix starts by splitting the matrix into 3 equal sub-matrices such that

$$\mathbf{X}_i = [\mathbf{X}_{i1}\ \mathbf{X}_{i2}\ \mathbf{X}_{i3}]^{\mathrm{T}} \tag{2}$$

This is followed by computing the accumulated differences (ADs) and statistical parameters (means and standard deviations) for each sub-matrix. The ADs and the statistical parameters are then concatenated together to form the final feature vector. The ADs of the three sub-matrices are computed as follows

$$\mathbf{d}_1 = \sum_{n=1}^{\frac{N}{3}} |\mathbf{x}_{n+1} - \mathbf{x}_n| \tag{3}$$

$$\mathbf{d}_2 = \sum_{n=\frac{N}{3}+1}^{\frac{2N}{3}} |\mathbf{x}_{n+1} - \mathbf{x}_n| \tag{4}$$

$$\mathbf{d}_3 = \sum_{n=\frac{2N}{3}+1}^{N} |\mathbf{x}_{n+1} - \mathbf{x}_n| \tag{5}$$

Additionally, the statistical parameters of the three sub-matrices are computed as follows

$$\boldsymbol{\mu}_j = E(\mathbf{X}_{ij});\ j = 1,2,3 \tag{6}$$

$$\boldsymbol{\sigma}_j = diag\left(Cov(\mathbf{X}_{ij})\right);\ \ j = 1,2,3 \tag{7}$$

Accordingly, the final feature vector for gesture *i* is

$$\mathbf{v}_i = [\mathbf{d}_1\ \ \boldsymbol{\mu}_1\ \ \boldsymbol{\sigma}_1\ \ \ \mathbf{d}_2\ \ \ \boldsymbol{\mu}_2\ \ \boldsymbol{\sigma}_2\ \ \ \mathbf{d}_3\ \ \boldsymbol{\mu}_3\ \ \boldsymbol{\sigma}_3] \tag{8}$$

Consequently, the final feature vector is comprised of 144 elements. It is worthwhile to mention that some of these elements might not offer discriminating information which may negatively impact the classification process. As such, we propose the use of stepwise regression to retain the discriminating elements in the feature vector.

3.2 Stepwise Regression

Stepwise regression is a wildly used regressor variable selection procedure. To illustrate the procedure (as described in [7]), assume that we have *K* candidate variables $x_1, x_2, \ldots, x_k$ and a single response variable *y*. In classification the candidate variables correspond to the elements of the feature vector and the response variable corresponds to the class label. Note that with the intercept term β_0, we end up with *K+1* variables.

In the procedure, the polynomial weights (or the regression model) are iteratively found by adding or removing variables at each step. The procedure starts by building a one variable regression model using the variable that has the highest correlation with the response variable *y*. This variable will also generate the largest partial F-statistic. In the second step, the remaining *K-1* variables are examined. The variable that generates the maximum partial F-statistic is added to the model provided that the partial F-statistic is larger than the value of the F-random variable for adding a variable to the model, such an F-random variable is referred to as fin. Formally the partial F-statistic for the second variable is computed by

$$f_2 = \frac{SS_R(\beta_2|\beta_1,\beta_0)}{MS_E(x_2,x_1)} \tag{9}$$

where $MS_E(x_2, x_1)$ denotes the mean square error for the model containing both x_1 and x_2. $SS_R(\beta 2|\beta_1,\beta_0)$ is the regression sum of squares due to β_2 given that β_1,β_0 are already in the model.

In general the partial F-statistic for variable j is computed by

$$f_j = \frac{SS_R(\beta_j|\beta_0,\beta_1,\ldots,\beta_{j-1},\beta_{j+1},\ldots,\beta_k)}{MS_E} \tag{10}$$

If variable x_2 is added to the model then the procedure determines whether the variable x_1 should be removed. This is determined by computing the F-statistic

$$f_1 = \frac{SS_R(\beta_1|\beta_2\beta_0)}{MS_E(x_2,x_2)} \tag{11}$$

If f_1 is less than the value of the F-random variable for removing variables from the model, such an F-random variable is referred to as f_{out}.

The procedure examines the remaining variables and stops when no other variable can be added or removed from the model. More information on stepwise regression can be found in classical statistics and probability texts such as [7].

It is also worth mentioning that one cannot arrive to the conclusion that all of the regressors that are important for predicting the response variable have been retained in the stepwise procedure. This is because such a procedure retains regressors based on the use of sample estimates of the true model weights. It is understood that there is a probability of making errors in retaining regressors.

3.3 Classification

In this paper we have used K-Nearest-Neighbor KNN classifier with Manhattan distance measure. For the user dependent mode, half the repetitions (i.e. 5 per gesture) were used as references while the remaining 5 were used for testing. This was done in round robin fashion such that all possible combinations for training and testing were considered (i.e. ${}^{10}C_5$ = 252 different combinations). On the other hand, for user independent recognition, we have used all the gesture repetitions of 5 users for training and the remaining gestures for the other 5 users for testing. Similar to the user dependent mode, all possible combinations of training and testing data were considered resulting in 252 different combinations.

It is worthwhile to mention that this classification technique is compared to the classical Dynamic Time Warping (DTW) [8] as applied to the sensor readings (i.e. without a feature extraction module).

4 Experimental Results

The following results are obtained using round robin on both the user independent and dependent modes. Using ADs and the Manhattan distance as a metric in a minimum distance classifier, we achieved recognition rates of 92.5% and 95.3% in the user independent and dependent modes respectively. Table 1 shows the confusion matrix for the recognition rates (%) of the 10 gestures.

Table 1. Resultant Confusion matrix using ADs and Manhattan distance

	G1	**G2**	**G4**	**G7**	**G8**	**G9**	**G10**	**G12**	**G13**	**G14**
G1	**97.7**	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00
G2	0.00	**89.3**	0.00	3.00	4.70	0.00	0.00	3.00	0.00	0.00
G4	2.00	0.00	**96.3**	1.70	0.00	0.00	0.00	0.00	0.00	0.00
G7	0.00	0.00	1.40	**92.6**	0.00	0.00	0.00	6.00	0.00	0.00
G8	0.00	3.00	0.00	0.00	**95.1**	0.00	0.01	0.00	1.80	0.00
G9	4.00	4.00	0.00	1.90	0.00	**90.1**	0.00	0.00	0.00	0.00
G10	0.00	0.00	0.10	6.00	0.00	0.00	**93.9**	0.00	0.00	0.00
G12	3.00	0.00	3.00	0.00	0.00	4.00	0.00	**88.1**	0.00	1.90
G13	5.80	0.00	0.80	0.00	0.00	0.00	0.00	0.00	**93.2**	0.20
G14	0.00	3.00	5.00	0.00	0.00	3.30	0.00	0.00	0.00	**88.7**

Table 2 shows the recognition results for the proposed scheme versus the classical DTW approach. Although the recognition results of the DTW are slightly higher, the proposed ADs solution is computationally less expensive. Namely, using the proposed scheme, classifying one gesture requires an average of 0.27 seconds. On the other hand, using DTW classical solution, the average time required to recognize a gesture is 4.9 seconds. Both times are measured on a dual core CPU running at 2.4 GHz.

Table 2. Recognition results

	User independent	User dependent
DTW	**95.1%**	**97.5%**
Ads	**92.5%**	**95.3%**

5 Conclusion

In this paper, we propose a low complexity Arabic sign language recognition system using sensor-based gloves. The solution is based on eliminating the time dependency of the data. The captured gesture sequence is projected into one or more static vectors. The dimensionality of the vectors is then reduced using stepwise regression. Classification rates of 92.5% and 95.3% for user independent and user dependent modes were achieved. As such, more elaborate classifiers for sequential data such as DTW are not needed for the classification task. Moreover, considering the complexity of the system, the proposed solution is simpler and faster. Hence, a real-time system can be implemented. The classification rates can be improved further on by extracting more features and giving different weights to different features.

Acknowledgments. The authors would like to acknowledge the Emirates Foundation (EF) for supporting this work via grant number EF #2009/031.

References

1. Shanableh, T., Assaleh, K.: Telescopic Vector Composition and Polar Accumulated Motion Residuals for Feature Extraction in Arabic Sign Language Recognition. EURASIP J. Image Video Process. 2007(87929), 1–10 (2007)
2. Chen, Y., Yang, J., Liou, S., Lee, G., Wang, J.: Online Classifier Construction Algorithm for Human Activity Detection Using a Tri-axial Accelerometer. Appl. Math. Comput. 205, 849–860 (2008)
3. Liu, J., Wang, Z., Zhong, L., Wickramasuriya, J., Vasudevan, V.: uWave: Accelerometer-based personalized gesture recognition and its applications. In: IEEE International Conference on Pervasive Computing and Communications, PerCom 2009, March 9-13, pp. 1–9 (2009)
4. Yang, J., Wang, J., Chen, Y.: Using Acceleration Measurements for Activity Recognition: An Effective Learning Algorithm for Constructing Neural Classifiers. Pattern Recogn. Lett. 29, 2213–2220 (2008)
5. Rezzoug, N., Gorce, P.: Prediction of Fingers Posture Using Artificial Neural Networks. J. Biomech. 41, 2743–2749 (2008)
6. DG Tech Engineering Solutions, `http://www.dg-tech.it/vhand/eng/`
7. Montgomery, D., Runger, G.: Applied Statistics and Probability for Engineers. John Wiley & Sons, U.S.A. (2010)
8. Holt, G.A., Reinders, M.J.T., Hendriks, E.A.: Multi-Dimensional Dynamic Time Warping for Gesture Recognition. Time 5249, 23–32 (2007)

Transductive Cartoon Retrieval by Multiple Hypergraph Learning

Jun Yu[1], Jun Cheng[2,3], Jianmin Wang[1], and Dacheng Tao[4]

[1] Computer Science Department, Xiamen University, Xiamen, 361005, China
[2] Shenzhen Institues of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[3] The Chinese University of Hong Kong, Shatin, Hong Kong,China
[4] Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia
he.zhang@aalto.fi

Abstract. Cartoon characters retrieval frequently suffers from the distance estimation problem. In this paper, a multiple hypergraph fusion based approach is presented to solve this problem. We build multiple hypergraphs on cartoon characters based on their features. In these hypergraphs, each vertex is a character, and an edge links to multiple vertices. In this way, the distance estimation between characters is avoided and the high-order relationship among characters can be explored. The experiments of retrieval are conducted on cartoon datasets, and the results demonstrate that the proposed approach can achieve better performance than state-of-the-arts methods.

Keywords: Retrieval, distance estimation, multiple, hypergraph fusion.

1 Introduction

Although cartoon is a popular and successful media in our life, its creation is usually of high-cost and labor-intensive, especially for the conventional 2D cartoon production. On the other hand, given the relatively long history of animation, there is a large-scale 'cartoon library' that consists of various animation materials. It is useful for animators to effectively create new animations by reusing.

Facing this opportunity and challenge, many attempts have been conducted in computer assisted animation, cartoon retrieval and synthesis [1][2] [12] [13] [15]. Juan et al. [1] offered a graph based cartoon synthesis approach that combines similar cartoon characters into a user directed sequence. Based on dissimilarity measure on cartoon characters' edges, this approach builds a graph [3] and generates a new cartoon sequence by finding the shortest path. It performs well for simple cartoon characters. But for characters with complex colors and motion patterns, it fails to generate smooth clips because edges can encode neither the color information nor the motion pattern. Yu et al. [2] proposed the simple-graph based transductive learning method for cartoon synthesis. The final cartoon synthesize is conducted as an iteration process where a group of similar cartoon

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 269–276, 2012.

characters is retrieved to form new sequences. However, this method also depends on edge feature. Thus, it cannot be used for complicated animations. To effectively and efficiently retrieve cartoons, it is essential to choose proper features for character representation. Inspired by [4] which indicates that cartoon characters are commonly represented by several features of different views, e.g., color, shape and motion, we introduce three visual features: color histogram [5], Hausdorff edge feature [6] and skeleton feature [7], to generate a complete and concise description of a cartoon character. These three kinds of features reveal different characteristics from distinct aspects and, at the same time, they are complementary to each other for a comprehensive representation.

Besides finding features to effectively represent cartoon characters, another critical issue is how to combine features of different views properly [14] in order to accurately measure the dissimilarity between characters. Though we can simply concatenate multiple features from different views together as a long vector, this solution is improper. That is because these features describe different aspects of a cartoon character's properties. Traditional learning methods [8][9] assume that the data are represented by a single feature. Thus, they cannot deal with data represented by multiple features. In [10], a multiview spectral embedding (MSE) algorithm is proposed to encode different features in different ways. Therefore, a physically meaningful embedding is achieved. MSE explores the complementary property of different views smoothly. However, in each view, MSE adopt the simple graph to estimate the data distribution. In this paper, we propose a novel cartoon character retrieval and recognition approach that explores the high-order relationship of cartoon character via hypergraphs. First, it adopts hypergraph to encode the representation for each feature. Here, each character is defined as a vertex and its several nearest neighbors form a specified edge. Thus, an edge can connect to multiple vertices for hypergraph. Besides, since each hyperedge connects to multiple characters, high-order information, such as whether three or more characters share close feature, can be explored.

2 Multiple Hypergraph Fusion Based Transductive Learning

In this section, we present the framework of the Multiple Hypergraph Fusion based Transductive Learning (MHF-TL) for cartoon retrieval. Fig. 1 presents the workflow of MHF-TL. First, a set of cartoon characters extracted from videos are imported as input. Then, multiple features of these characters including color histogram (CH), Hausdorff edge feature (HEF) and skeleton feature (SF) are imported as input. Subsequently, the hypergraph model is adopted to construct the hypergraph Laplacian which can estimate the distribution of the characters efficiently. Afterward, these hypergraph Laplacian can be fused linearly by using a group of weights. Meanwhile, the user can add the label information into this framework. Thus, a group of optimal weights can be obtained by using the Multi-Hypergraph Fusion based Transductive Learning. Finally, the obtained value matrix can be used in cartoon retrieval.

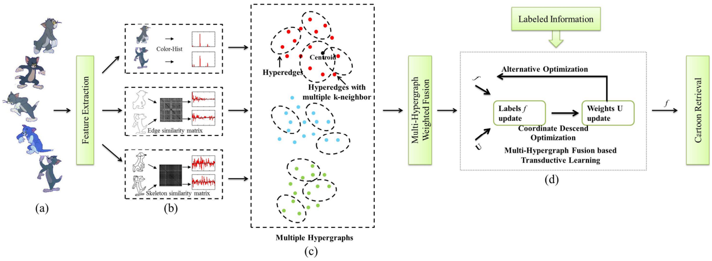


Fig. 1. Workflow of Multiple Hypergraph Fusion based Transductive Learning (MHF-TL) for Cartoon Retrieval (a) Input of data; (b) multiple features extraction for character representation; (c) hypergraph construction for each feature (d) Multiple-Hypergraph Fusion based Transductive Learning

2.1 Hypergraph Construction

Now we introduce the approach of hypergraph construction in cartoon retrieval. We regard each character in the database as a vertex in the hypergraph $G = (V, E, w)$. Let $V = v_1, v_2, \ldots, v_n$ represent n vertices, and $E = e_1, e_2, \ldots, e_m$ represent m hyperedges. For each hyperedge, there is an associated positive number $w(e)$, named the value of hyperedge e. In our approach, a hyperedge is constructed from a centroid vertex and its related k neighbors. Thus, for each feature, we can construct a series of hypergraph by coordinating the k neighbors in a specified range. To calculate the weight for each hyperedge $w(e)$, we first calculate a $|V| \times |V|$ affinity matrix A over V.

$$A(i, j) = exp(-\frac{\|v_i - v_j\|}{D}), \tag{1}$$

where D is the average distance among the vertices and $A(i, j) \in [0, 1]$. Then, the incidence matrix H of the hypergraph can be calculated as:

$$h(v_i, e_j) = \{ \begin{matrix} A(j, i) \text{ if } v_i \in e_j \\ 0, \quad \text{otherwise.} \end{matrix} \tag{2}$$

In accordance with this formulation, the vertex v_i is assigned into e_j, in which v_j is the centroid of e_j. In this way, the correlation information among vertices can be accurately described. Hence, the hyperedge weight can be calculated as:

$$w(e_i) = \sum_{v_j \in e_i} A(i, j). \tag{3}$$

Based on the definition of weights, the degree $d(v)$ of a vertex $v \in V$ and the degree $\delta(e)$ of the hyperedge $e \in E$ can be calculated as:

$$d(v) = \sum_{e \in E} w(e)h(v, e), \tag{4}$$

and

$$\delta(e) = \sum_{v \in V} h(v, e). \tag{5}$$

In our approach, the hypergraph model with variable k-nearest neighbors is efficient in describing the data distribution. When the distribution of data is dense, the parameter of k-nearest neighbor tends toward a small value. When the distribution of data is sparse, the parameter k inclines toward the small value.

2.2 Multiple Hypergraph Fusion Based Cartoon Retrieval

As mentioned in [11], different machine learning tasks can be conducted on hypergraphs. Thus, the transductive learning framework can be formulated as:

$$\arg\min_{f}\{\varOmega(f) + \lambda R_{emp}(f)\}, \tag{6}$$

where f is the classification function to be learned, $\varOmega(f)$ is a regularizer on the hypergraph, $R_{emp}(f)$ is empirical loss, and $\lambda > 0$ is a tuning parameter. According to [11], the regularizer on the hypergraph is defined as:

$$\varOmega(f) = \frac{1}{2}\varSigma_{e \in E}\varSigma_{u,v \in V}\frac{w(e)h(u,e)h(v,e)}{\delta(e)}(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}})^2. \tag{7}$$

Let $\varTheta = D_v^{-\frac{1}{2}}HWD_e^{-1}H^TD_v^{-\frac{1}{2}}$, and $L = I - \varTheta$, the normalized cost function can be written as:

$$\varOmega(f) = f^TLf, \tag{8}$$

where L is a positive semi-definite matrix, and it is called hypergraph Laplacian. Besides, the loss term can be defined as follows:

$$\|f - y\|^2 = \varSigma_{u \in V}(f(u) - y(u))^2, \tag{9}$$

where y is a label vector. It can be assumed that the number of character in cartoon database is n, and the ith character is chosen as the query sample. Thus, y is denoted by an $n \times 1$ vector, where each element of y is 0 except its ith value is 1. Thus, the learning task is to minimize the sum of two terms

$$\arg\min_{f}(f^TLf + \lambda\|f - y\|^2. \tag{10}$$

As mentioned in Section 2.1, the construction of the hypergraph laplacian involves setting the k-neighbor. Besides, varied hypergraph laplacian matrix can be constructed for each feature. In order to automatically, effectively and efficiently approximate the optimal hypergraph Laplacian, we adopt the alternative approach to learn the optimal Laplacian implicitly. Since the optimal hypergraph

Laplacian is the discrete approximation to the manifold[16], the above assumption is equivalent to constraining the search space of possible graph Laplacians, i.e.,

$$\begin{array}{c} L = \Sigma_{i=1}^{N}\mu_i L_i \\ s.t.\Sigma_{i=1}^{N}\mu_i = 1, \mu_i > 0, i = 1, \ldots, N \end{array}, \tag{11}$$

where N represents all possible hypergraph Laplacian.

Under this constraint, the optimal hypergraph Laplacian estimation is turned to the problem of learning the optimal linear combination of some pre-given candidates. Then, the multiple hypergraph fusion based transductive learning can be formulated as

$$\begin{array}{c} f^T(\sum_{i=1}^{N}\mu_i L_i)f + \lambda\|f-y\|^2 + \beta\|\mu\|^2 \\ s.t.\Sigma_{i=1}^{N}\mu_i = 1, \mu_i > 0, i = 1, \ldots, N \end{array}, \tag{12}$$

where the regularization term $\|\mu\|^2$ is adopted to avoid the overfitting to one manifold, and β is the trade-off parameter to control the contribution of the regularization term $\|\mu\|^2$. For a fixed μ, Eq. (12) degenerates to Eq. (10) with $L = \sum_{i=1}^{N}\mu_i L_i$. On the other hand, for a fixed f, Eq. (12) can be simplified to

$$\begin{array}{c} \min_{\mu \in R^N} \sum_{i=1}^{N}\mu_i p_i + \beta\|\mu\|^2, \\ s.t. \sum_{i=1}^{N}\mu_i = 1, \mu_i > 0, i = 1, \ldots, N \end{array}, \tag{13}$$

where $p_i = f^T L_i f$. Theoretically, the hypergraph Laplacian matrix of each graph satisfies the semidefinite positive property. Thus, their linear combination is also semidefinite positive. To obtain the solution of Eq. (13), the alternating optimization is implemented with a fixed μ, as well as the solution μ with a fixed f.

To fix μ, we get the partial derivative with respect to f, and we can obtain:

$$\begin{array}{c} \frac{\partial}{\partial f}[f^T(I - \sum_{i=1}^{N}\mu_i(D_v^i)^{-\frac{1}{2}}U^i W^i (H^i)^T (D_v^i)^{-\frac{1}{2}})f + \lambda\|f-y\|^2] = 0 \\ \Rightarrow f = \frac{\lambda}{1+\lambda}(I - \frac{\Sigma_{i=1}^{N}\mu_i S^i}{1+\lambda})^{-1}y \end{array} \tag{14}$$

where $S^i = (D_v^i)^{-\frac{1}{2}}U^i W^i (H^i)(D_v^i)^{-\frac{1}{2}}$. U^i represents hyperedges' degree $(D_e^i)^{-1}$ with the elements in diagonal $(U_1^i, (U_2^i, ..., (U_m^i)$. On the other side, to learn μ with a fixed f, we obtain Eq. (13). In this case, the coordinate descend algorithm can be adopted. In each iteration, two elements μ_i and μ_j are selected for updating while the others are fixed. Due to the constraint $\sum_{i=1}^{N}\mu_i = 1$, the summation of μ_i and μ_j will not change after this iteration. Hence, the solution can be obtained as:

$$\begin{array}{c} \mu_i^* = 0,\ \mu_j^* = \mu_i + \mu_j, \text{ if } 2\beta(\mu_i + \mu_j) + (p_j - p_i) \le 0 \\ \mu_i^* = \mu_i + \mu_j,\ \mu_j^* = 0, \text{ if } 2\beta(\mu_i + \mu_j) + (p_i - p_j) \le 0 \\ \mu_i^* = \frac{2r(\mu_i + \mu_j) + (p_j - p_i)}{4\beta},\ \mu_j^* = \mu_i + \mu_j - \mu_i^*, \text{ else} \end{array} \tag{15}$$

The coordinate descend method is adopted to iteratively traverse over all pairs of elements in μ and the solution in Eq. (14) is adopted until the objective function in Eq. (13) does not decrease.

3 Experiments

3.1 Datasets

To evaluate the performance of the proposed method, we collected cartoon characters of TOM and JERRY from cartoon videos. The data are classified into two character groups. In order to annotate cartoon characters, cartoonists in our lab classified the characters with similar color, shape and motion into one category. For each cartoon group, we obtained 50 categories of characters. The number of characters in each category is 10. In this case, 1,000 cartoon characters for this experiment are collected. Fig. 2 shows some samples in this database.

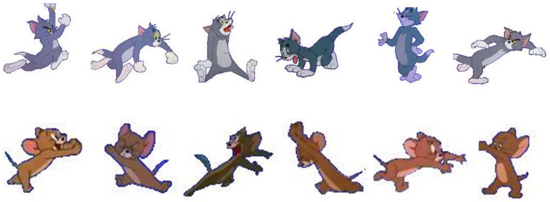


Fig. 2. Cartoon character samples for TOM and JERRY in the datasets

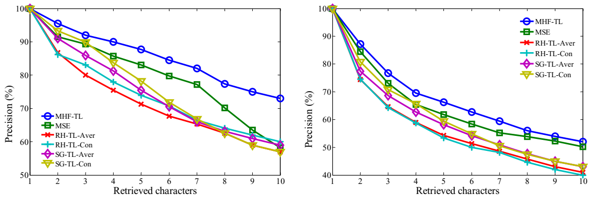
3.2 Experimental Configurations

In this experiment, each class has 10 characters. Here, each character is used as query character. For retrieval, the most similar m characters (retrieved characters) are obtained, among which the number of characters from the same class (relevant characters) as the query character is recorded. Thus, we adopt the Precision Curve to evaluate the performance. We compare the proposed multiple hypergraph fusion based transductive learning (MHF-TL) with Multiview Spectral Embedding (MSE) [10], Regular Hypergraph based Transductive Learning (RH-TL) and simple-graph based transductive learning (SG-TL). Specifically, in MHF-TL, the range of k neighbors for each centroid vertex is fixed as [3,5,10,15,20]. Both RH-TL and SG-TL can be conducted with the features of CH, HEF and SF. Thus, we can obtain the average results of RH-TL-Aver and SG-TL-Aver. In addition, we can conduct RH-TL and SG-TL by concatenating the three features into a long vector as RH-TL-Con and SG-TL-Con.

3.3 Experimental Results

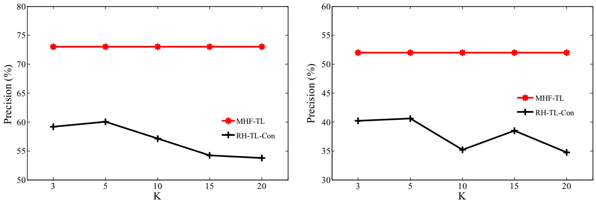
Fig. 3 records the precision curves on the datasets of TOM and JERRY. The results show that the proposed Multiple Hypergraph Fusion based Transductive Learning (MHF-TL) achieves better retrieval results than all other methods.

We then observe the retrieval performance of using different individual hypergraphs. Figure 4 illustrates that the performance comparison of using individual



(a) TOM (b) JERRY

Fig. 3. The precision curve comparison of MHF-TL, MSE, RH-TL-Aver, RH-TL-Con, SG-TL-Aver and SG-TL-Con. (a) results on the TOM dataset; (b) results on the JERRY dataset; (c) results on the Squatters dataset.



(a) TOM (b) JERRY

Fig. 4. Retrieval comparison of using RH-TL-Con with different K and using MHF-TL. (a) results on the TOM dataset; (b) results on the JERRY dataset.

hypergraphs generated with different k (varied in the range [3,5,10,15,20]) and using fused hypergraph. We can see that the performance curves mainly exhibit a ” or ” shape when k varies. This can be explained that the discriminative ability of hypergraph will be weak when k is too small and many characters will not be connected when k is too large. This makes our method robust and feasible as we do not need to tune the parameter k.

4 Conclusions

In this paper, we propose a novel cartoon character retrieval using constructive hypergraph analysis. Multiple hypergraphs at different granularities are constructed based on the features of cartoon characters. These hypergraphs comprehensively capture the high-order relationship among cartoon characters and the scheme avoids the pairwise distance estimation. Experimental results have clearly demonstrated the superiority of the hypergraph-based approach.

Acknowledgments. This work is supported by National Natural Science Foundation of China (No. 61100104), the Natural Science Foundation of Fujian Province of China (No. 2012J01287), CAS and Locality Cooperation

Projects (ZNGZ-2011-012), Shenzhen-Kong Kong Innovation Circle Project (JSE201007200037A), Guangdong-CAS Strategic Cooperation Program (2010A090100016), Introduced Innovative RD Team of Guangdong Province ’Robot and Intelligent Information Technology’, Shenzhen Technology Project (ZD200904300074A) and Guangdong-Hongkong Technology Cooperation Funding(2011A091200001).

References

1. Juan, D., Bodenheimer, B.: Cartoon Textures. In: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 267–276 (2004)
2. Yu, J., Liu, D., Seah, H.: Transductive Graph based Cartoon Synthesis. Computer Animation and Virtual Worlds 21(3), 277–288 (2010)
3. Tenenbaum, J., Silva, V., Langford, J.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290(22), 2319–2323 (2000)
4. Wang, J., Drucker, S., Agrawala, M., Cohen, M.: The Cartoon Animation Filter. In: ACM Int. Conf. Computer Graphics and Interactive Techniques, pp. 1169–1173 (2006)
5. Shapiro, L., Stockman, G.: Computer Vision. Prentice Hall (2003)
6. Hutten, D., Locker, G., Ruchlidge, W.: Comparing Images Using the Hausdorff Distance. IEEE Trans. Pattern Anal. Mach. Intell. 15(9), 850–863 (1993)
7. Bai, X., Latecki, L., Liu, W.: Skeleton pruning by contour partitioning with discrete curve evolution. IEEE Trans. Pattern Anal. Mach. Intell. 29(3), 449–462 (2007)
8. Belhumeur, P., Hepanha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. Pattern Anal. Mach. Intell. 19(7), 711–720 (1997)
9. Belkin, M., Niyogi, P.: Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In: Neural Information Processing Systems, pp. 585–591 (2002)
10. Xia, T., Tao, D., Mei, T., Zhang, Y.: Multiview Spectral Embedding. IEEE Trans. Syst. Man and Cybernetics. Part II 40(6), 1438–1446 (2010)
11. Zhou, D., Huang, J., Schölkopf, B.: Learning with hypergraphs: Clustering, classification, and embedding. In: Neural Information Processing Systems, pp. 1601–1608 (2006)
12. Yu, J., Liu, D., Tao, D., Seah, H.: Complex Object Correspondence Construction in 2D Animation. IEEE Trans. Image Processing 20(11), 3257–3269 (2011)
13. Yu, J., Liu, D., Tao, D., Seah, H.: On Combining Multiple Features for Cartoon Character Retrieval and Clip Synthesis. IEEE Trans. Syst. Man and Cybernetics, Part II (2012), doi:10.1109/TSMCB
14. Xie, B., Mu, Y., Tao, D., Huang, K.: m-SNE: Multiview Stochastic Neighbor Embedding. IEEE Transactions on Systems, Man, and Cybernetics, Part B 41(4), 1088–1096 (2011)
15. Tian, X., Tao, D., Rui, Y.: Sparse Transfer Learning for Interactive Video Search Reranking. ACM TOMCCAP (2012)
16. Guan, N., Tao, D., Luo, Z., Yuan, B.: Non-Negative Patch Alignment Framework. IEEE Transactions on Neural Networks 22(8), 1218–1230 (2011)

Adaptive Multiplicative Updates for Projective Nonnegative Matrix Factorization*

He Zhang, Zhirong Yang, and Erkki Oja

Department of Information and Computer Science
Aalto University School of Science, Espoo, Finland
{he.zhang,zhirong.yang,erkki.oja}@aalto.fi

Abstract. Projective Nonnegative Matrix Factorization (PNMF) is able to extract sparse features and provide good approximation for discrete problems such as clustering. However, the original PNMF optimization algorithm can not guarantee theoretical convergence during the iterative learning. We propose here an adaptive multiplicative algorithm for PNMF which is not only theoretically convergent but also significantly faster than the previous implementation. An adaptive exponent scheme has been adopted for our method instead of the old unitary one, which ensures the theoretical convergence and accelerates the convergence speed thanks to the adaptive exponent. We provide new multiplicative update rules for PNMF based on the squared Euclidean distance and the I-divergence. For the empirical contributions, we first provide a counter example on the monotonicity using the original PNMF algorithm, and then verify our proposed method by experiments on a variety of real-world data sets.

Keywords: Adaptive, multiplicative updates, PNMF, NMF.

1 Introduction

Recently *Nonnegative Matrix Factorization* (NMF) has been attracting much research effort and applied to many different fields such as face recognition, document clustering, gene expression studies, music analysis [7,1,3]. The research stream originates from the work by Lee and Seung [6], in which they showed that the nonnegativity constraint and the related multiplicative update rules can generate part-based representations of the data. However, the sparseness achieved by NMF is only mediocre. Many NMF variants (e.g. [4,5]) addressed this problem but their solutions often require extra user-specified parameters to achieve sparser results, which is inconvenient in practice.

Projective Nonnegative Matrix Factorization (PNMF) [10,9] as a new variant of NMF has shown advantages over NMF in learning a sparse or orthogonal factorizing matrix, which is desired in both feature extraction and clustering. Typically PNMF follows the NMF optimization approach by using multiplicative

[*] Supported by the Academy of Finland in the project Finnish Center of Excellence in Computational Inference Research (COIN).

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 277–284, 2012.

updates. However, the original PNMF algorithm does not guarantee monotonic decrease of the dissimilarity between the input matrix and its approximate after each learning iteration.

We propose new multiplicative algorithms for PNMF in this paper. The convergence problem of the original PNMF update rules is caused by the restrict that the exponent in the update rule must be unitary (i.e., one). Dropping the restrict, we can obtain theoretically convergent update rules without extra normalization steps. The multiplicative updates are further relaxed by allowing variable exponents in different iterations, which turns out to be an effective strategy for accelerating the optimization. The failure of the original PNMF algorithm is demonstrated by an counter example, where the monotonicity of the objective evolution is violated. By contrast, our new method steadily minimizes the objective and converges significantly faster than the old algorithms.

In the remaining, Section 2 recapitulates the essence of the PNMF objectives and their previous optimization methods. Section 3 presents the new convergent multiplicative update rules and the fast PNMF algorithm by using adaptive exponents. In Section 4, we empirically compared the proposed methods using a variety of data sets, and Section 5 concludes the paper.

2 Projective Nonnegative Matrix Factorization

Given a nonnegative data matrix $\mathbf{X} \in \mathbb{R}_+^{m \times n}$, *Projective Nonnegative Matrix Factorization* (PNMF) seeks a decomposition of $\mathbf{X}$ that is of the form: $\mathbf{X} \approx \mathbf{W}\mathbf{W}^T\mathbf{X}$, where $\mathbf{W} \in \mathbb{R}_+^{m \times r}$ with the rank $r < \min(m, n)$. Compared with the NMF approximating scheme $\mathbf{X} \approx \mathbf{W}\mathbf{H}$, PNMF replaces $\mathbf{H}$ with $\mathbf{W}^T\mathbf{X}$. This replacement has shown to have positive consequence in sparseness of the approximation, orthogonality of the factorizing matrix, close equivalence to clustering, generalization of the approximation to new data without heavy re-computations, and easy extension to a nonlinear kernel method [9].

Let $\widehat{\mathbf{X}} = \mathbf{W}\mathbf{W}^T\mathbf{X}$ denote the approximating matrix. The approximation can be achieved by minimizing two widely used objectives: (i) the *Squared Euclidean distance* (Frobenius norm) defined as $D_{\mathrm{EU}}\left(\mathbf{X}||\widehat{\mathbf{X}}\right) = \sum_{ij}\left(X_{ij} - \widehat{X}_{ij}\right)^2$, and (ii) the *Non-normalized Kullback-Leibler divergence* (I-divergence) defined as $D_{\mathrm{I}}\left(\mathbf{X}||\widehat{\mathbf{X}}\right) = \sum_{ij}\left(X_{ij}\log\frac{X_{ij}}{\widehat{X}_{ij}} - X_{ij} + \widehat{X}_{ij}\right)$. Note that PNMF is also called Clustering NMF which was later proposed by Ding et al. in [2].

Denote $Z_{ij} = X_{ij}/\widehat{X}_{ij}$ and $\mathbf{1}_m$ a column vector of length m filled with 1. To minimize the above objectives, the authors in [10,9] have employed the following multiplicative update algorithms:

$$W'_{ik} = W_{ik}\frac{(\mathbf{AW})_{ik}}{(\mathbf{BW})_{ik}}, \tag{1}$$

where for the Euclidean case

$$\mathbf{A} = 2\mathbf{X}\mathbf{X}^T \text{ and } \mathbf{B} = \mathbf{W}\mathbf{W}\mathbf{X}\mathbf{X}^T + \mathbf{X}\mathbf{X}^T\mathbf{W}\mathbf{W}^T, \tag{2}$$

and for the I-divergence case

$$\mathbf{A} = \mathbf{Z}\mathbf{X}^T + \mathbf{X}\mathbf{Z}^T \text{ and } \mathbf{B} = \mathbf{1}_m\mathbf{1}_n^T\mathbf{X}^T + \mathbf{X}\mathbf{1}_n\mathbf{1}_m^T. \tag{3}$$

Note that the update rule (1) itself does not necessarily decrease the objective in each iteration and must therefore be accompanied with a normalization or stabilization step, i.e.,

$$\mathbf{W}^{\text{new}} = \mathbf{W}'/\|\mathbf{W}'\|, \tag{4}$$

where $\|\mathbf{W}'\|$ equals the square root of maximal eigenvalue of $\mathbf{W}'^T\mathbf{W}'$. Though the algorithms using the update rules (1) and (4) usually work in practice, the theoretical proof of their convergence is still lacking. In Section 4.1 we can even provide a counter example of these rules for the I-divergence.

3 Adaptive PNMF

The derivation of the update rule (1) follows a heuristic principle that puts the unsigned negative terms in the gradient to the numerator and the rest to the denominator of the multiplying factor to $\mathbf{W}$. Update rules obtained by this principle may not decrease the objective at each iteration [9] because the exponent of the multiplying factor is restricted to one. Discarding the restrict, we can obtain theoretically convergent multiplicative update rules in the relaxed form

$$W_{ik}^{\text{new}} = W_{ik}\left[\frac{(\mathbf{AW})_{ik}}{(\mathbf{BW})_{ik}}\right]^{\eta} \tag{5}$$

where $\eta \in \mathbb{R}_+$ and the convergence is guaranteed by the following theorem.

Theorem 1. *The multiplicative update (5) monotonically decrease $D_{EU}\left(\mathbf{X}||\widehat{\mathbf{X}}\right)$ with $\eta = 1/3$, and decrease $D_I\left(\mathbf{X}||\widehat{\mathbf{X}}\right)$ with $\eta = 1/2$.*

The proof is special cases of Majorization-Minimization development procedure in [8]. For self-contained purpose, we include the proof sketch in the Appendix.

The multiplicative algorithm using the update rule (5) avoids unwanted rises and thus assures theoretical convergence of the iterative learning. However, the exponent η that remains constant throughout the iterations is often conservative in practice. Here we propose to accelerate the learning by using more aggressive choice of the exponent, which adaptively changes during the iterations.

A simple strategy is to increase the exponent steadily if the new objective is smaller than the old one and otherwise shrink back to the safe choice, η. The pseudo-codes for such implementation is given in Algorithm 1, where $D(\mathbf{X}||\widehat{\mathbf{X}})$, $\mathbf{A}$, $\mathbf{B}$ and η are defined according to the type of cost function (Euclidean distance or I-divergence). We have empirically used $\mu = 0.1$ in all related experiments in this work. Although more comprehensive adaptation approaches could be applied, we find that such a simple strategy can already significantly speed up the convergence while still maintaining the monotonicity of updates.

Algorithm 1. Multiplicative Updates with Adaptive Exponent for PNMF

```
Usage: $\mathbf{W} \leftarrow \mathrm{FastPNMF}(\mathbf{X}, \eta, \mu)$.
Initialize $\mathbf{W}$; $\rho \leftarrow \eta$.
repeat
    $U_{ik} \leftarrow W_{ik} \left[ \frac{(\mathbf{AW})_{ik}}{(\mathbf{BW})_{ik}} \right]^{\rho}$
    if $D(\mathbf{X}||\mathbf{UU}^T\mathbf{X}) < D(\mathbf{X}||\mathbf{WW}^T\mathbf{X})$ then
        $\mathbf{W} \leftarrow \mathbf{U}$
        $\rho \leftarrow \rho + \mu$
    else
        $\rho \leftarrow \eta$
    end if
until convergent conditions are satisfied
```

4 Experiments

We have selected a variety of data sets that are commonly used in machine learning for our experiments. These data sets were obtained from the UCI repository[1], the University of Florida Sparse Matrix Collection[2], and the LSI text corpora[3], as well as other publicly available websites. The statistics of the data sets are summarized in Table 1.

Table 1. The data sets used in the experiments (m = Dimensions, n = # of samples)

	GD95_b	wine	sonar	mfeat	orl	feret	worldcities	swimmer	cisi	cran	med
m	40	13	60	292	400	1024	313	256	1460	1398	1033
n	69	178	208	2000	10304	2409	100	1024	5609	4612	5831

For the empirical comparisons, we consider three methods: (i) *PNMFn*, i.e, the original PNMF algorithm using the multiplicative update rule (1) and the *n̲ormalization* step (4), (ii) *PNMFc*, i.e., the convergent multiplicative PNMF algorithm (5) using *c̲onstant* exponent according to Theorem 1, and (iii) *PNMFa*, i.e., the fast adaptive PNMF algorithm using *a̲daptive* exponents (Algorithm 1).

4.1 A Counter-Example of Using Extra Normalization

Figure 1 shows a counter-example of the original PNMF algorithm for I-divergence using Eqs. (1), (3), and (4). We have used the *GD95_b* data set in the experiment. It can be seen that the monotonicity of objective evolution is violated in every other loop since the 19th iteration and the optimization is then stuck in an endless fluctuation without a decreasing trend.

[1] http://archive.ics.uci.edu/ml/

[2] http://www.cise.ufl.edu/research/sparse/matrices/index.html

[3] http://www.cs.utk.edu/~lsi/corpa.html

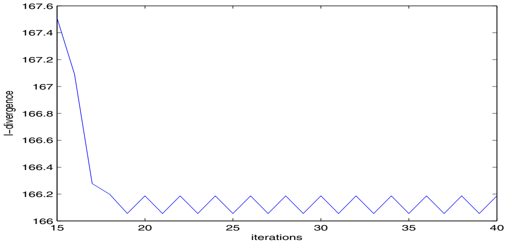


Fig. 1. A counter example showing that the original PNMF algorithm with normalization does not monotonically decrease the I-divergence for the *GD95_b* data set

4.2 Training Time Comparison

Figure 2 shows the objective evolution curves using the compared methods. One can see that the objectives of the proposed methods, PNMFc and PNMFa, monotonically decrease for the whole iterative learning process without any unexpected rises. Furthermore, PNMFa generally converges the fastest as its curves are below the other two in all plots.

In addition to qualitative analysis, we have also compared the benchmark on converged time of the three methods. Table 2 summarizes the means and standard deviations of the resulting converged time. The converged time is calculated as follows. We first find the earliest iteration of PNMFn where the objective D_n is sufficiently close to its minimum D^*: $|D_n - D^*|/D^* < 0.001$. The corresponding time is recorded as the converged time of the PNMFn. For the PNMFc evolution, the converged time is that of the first iteration where the objective D_c fulfills $|D_c - D^*|/D^* < 0.001$. If no such iteration exists, the converged time of PNMFc is set to the largest learning time of the three methods. The same procedure of *PNMFc* is applied to *PNMFa*. Each algorithm on each dataset has been repeated 100 times with different random seeds for initialization. These quantitative results confirm that PNMFa is the fastest algorithm among the

Table 2. The mean (μ) and standard deviation (σ) of the converged time (seconds)

(a) Criterion: the squared Euclidean distance

method	wine	sonar	mfeat	orl	feret
PNMFn	0.97±0.03	0.97±0.01	26.14±1.54	40.37±1.03	30.80±7.34
PNMFc	0.22±0.11	0.22±0.11	68.57±1.75	117.26±1.74	107.58±24.43
PNMFa	0.06±0.03	0.06±0.03	19.10±0.70	29.89±1.48	19.97±5.60

(b) Criterion: the I-divergence

method	worldcities	swimmer	cisi	cran	med
PNMFn	8.35±3.71	309.78±8.78	478.43±43.51	438.98±41.71	321.94±34.90
PNMFc	14.07±2.98	613.04±20.63	863.89±69.23	809.61±62.64	566.99±64.44
PNMFa	4.68±1.44	193.47±5.43	193.23±18.70	189.41±18.50	132.67±13.86

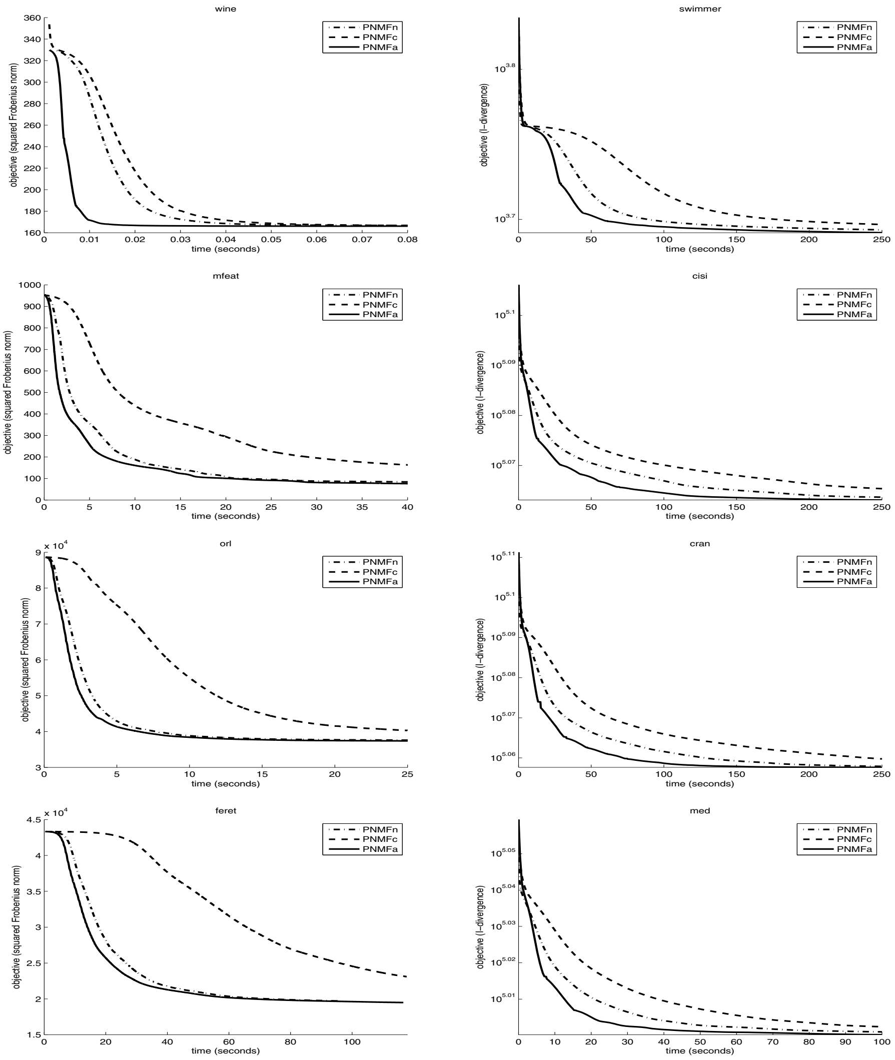


Fig. 2. Evolutions of objectives using the compared methods based on (left) squared Euclidean distance and (right) I-divergence

three compared: it is 1.5 to 2 times faster than PNMFn and 3 to 5 times faster than PNMFc. The advantage over PNMFn is more significant for the two small data sets *wine* and *sonar*.

5 Conclusions

We have proposed a fast multiplicative algorithm for Projective Nonnegative Matrix Factorization. Our method gets rid of two restricts in the conventional multiplicative update rules. Firstly, relaxing the exponent of the multiplying factor to any positive number can lead to theoretically convergent update rules without extra normalization. Secondly, further relaxation by allowing variable exponent can accelerate the iterative learning. Empirical results show that the proposed algorithm not only monotonically decreases the dissimilarity objective but also converges significantly faster than the previous implementation.

The accelerated algorithms facilitate application of the PNMF method. More large-scale datasets will be tested in the future. Moreover, the proposed adaptive exponent technique is readily extended to other fixed-point algorithms that use multiplicative updates.

A Appendix: Proof of Theorem 1

A.1 The Euclidean Distance Case

We rewrite the squared Euclidean distance as

$$D_{\mathrm{EU}}(\mathbf{X}||\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\mathbf{X}) = -2\mathrm{Tr}\left(\widetilde{\mathbf{W}}^T\mathbf{X}\mathbf{X}^T\widetilde{\mathbf{W}}\right) + \sum_{ij}\left(\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\mathbf{X}\right)^2_{ij} + \mathrm{constant}. \quad (6)$$

The first term on the right is upper-bounded by its linear expansion at the current estimate $\mathbf{W}$:

$$-2\mathrm{Tr}\left(\widetilde{\mathbf{W}}^T\mathbf{X}\mathbf{X}^T\widetilde{\mathbf{W}}\right) \leq -4\sum_{ik}\widetilde{W}_{ik}\left(\mathbf{X}\mathbf{X}^T\mathbf{W}\right)_{ik} + \mathrm{constant} \quad (7)$$

because it is concave with respect to $\widetilde{\mathbf{W}}$. Next, let $\lambda_{ijak} = \frac{W_{ik}W_{ak}X_{aj}}{(\mathbf{W}\mathbf{W}^T\mathbf{X})_{ij}}$. The second term can be upper-bounded by using Jensen's inequality as follows:

$$\sum_{ij}\left(\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\mathbf{X}\right)^2_{ij} \leq \frac{\widetilde{W}^4_{ik}}{2W^3_{ik}}\left(\mathbf{W}\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W} + \mathbf{X}\mathbf{X}^T\mathbf{W}\mathbf{W}^T\mathbf{W}\right)_{ik}. \quad (8)$$

We can then construct the auxiliary function

$$G(\widetilde{\mathbf{W}},\mathbf{W}) = -2\mathrm{Tr}\left(\widetilde{\mathbf{W}}^T\mathbf{A}\mathbf{W}\right) + \sum_{ik}\frac{\widetilde{W}^4_{ik}}{2W^3_{ik}}\left(\mathbf{B}\mathbf{W}\right)_{ik} + \mathrm{constant} \quad (9)$$

which upper bounds $D_{\mathrm{EU}}(\mathbf{X}||\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\mathbf{X})$ with $\mathbf{A}$ and $\mathbf{B}$ defined in Eq. (2). Minimizing $G(\widetilde{\mathbf{W}},\mathbf{W})$ is implemented by setting its gradient with respect to $\widetilde{\mathbf{W}}$ to zero, which yields the update rule Eq. (5) with $\eta = 1/3$.

A.2 The I-Divergence Case

We rewrite the I-divergence as

$$D_{\mathrm{I}}(\mathbf{X}||\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\mathbf{X}) = -\sum_{ij}X_{ij}\log\left(\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\mathbf{X}\right)_{ij} + \sum_{ij}\left(\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\mathbf{X}\right)_{ij} + \mathrm{constant}. \quad (10)$$

The first term is upper-bounded using the Jensen's inequality:

$$-\sum_{ij} X_{ij} \log\left(\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\mathbf{X}\right)_{ij} \leq -\sum_{aik} A_{ai} W_{ik} W_{ak} \log\left(\widetilde{W}_{ik}\widetilde{W}_{ak}\right) + \text{constant}, \tag{11}$$

where **A** is defined in Eq. (3). For the second term, we can rewrite it with **B** defined in Eq. (3) and obtain its upper bound with $\widetilde{U}_{ik} = \widetilde{W}_{ik}$ and $U_{ik} = W_{ik}$.

$$\sum_{ij}\left(\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\mathbf{X}\right)_{ij} = \frac{1}{2}\mathrm{Tr}\left(\widetilde{\mathbf{W}}^T\mathbf{B}\widetilde{\mathbf{W}}\right) \leq \sum_{ik} \frac{\widetilde{W}_{ik}^2}{2W_{ik}} (\mathbf{BW})_{ik} \tag{12}$$

We can then construct the auxiliary function

$$G(\widetilde{\mathbf{W}}, \mathbf{W}) = -\sum_{aik} A_{ai} W_{ik} W_{ak} \log\left(\widetilde{W}_{ik}\widetilde{W}_{ak}\right) + \sum_{ik} \frac{\widetilde{W}_{ik}^2}{2W_{ik}} (\mathbf{BW})_{ik} + \text{constant}, \tag{13}$$

which upper bounds $D_{\mathrm{I}}(\mathbf{X}||\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\mathbf{X})$. Setting the gradient of $G(\widetilde{\mathbf{W}}, \mathbf{W})$ with respect to $\widetilde{\mathbf{W}}$ to zero, we obtain the update rule (5) with $\eta = 1/2$.

References

1. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis. John Wiley (2009)
2. Ding, C., Li, T., Jordan, M.: Convex and semi-nonnegative matrix factorizations. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(1), 45–55 (2010)
3. Févotte, C., Bertin, N., Durrieu, J.L.: Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis. Neural Computation 21(3), 793–830 (2009)
4. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research 5, 1457–1469 (2004)
5. Kim, H., Park, H.: Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics 23(12), 1495–1502 (2007)
6. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
7. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 267–273 (2003)
8. Yang, Z., Oja, E.: Unified development of multiplicative algorithms for linear and quadratic nonnegative matrix factorization. IEEE Transactions on Neural Networks 22(12), 1878–1891 (2011)
9. Yang, Z., Oja, E.: Linear and nonlinear projective nonnegative matrix factorization. IEEE Transaction on Neural Networks 21(5), 734–749 (2010)
10. Yuan, Z., Oja, E.: Projective nonnegative matrix factorization for image compression and feature extraction. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 333–342. Springer, Heidelberg (2005)

Online Projective Nonnegative Matrix Factorization for Large Datasets*

Zhirong Yang, He Zhang, and Erkki Oja

Department of Information and Computer Science
Aalto University School of Science, Espoo, Finland
{zhirong.yang,he.zhang,erkki.oja}@aalto.fi

Abstract. Projective Nonnegative Matrix Factorization (PNMF) is one of the recent methods for computing low-rank approximations to data matrices. It is advantageous in many practical application domains such as clustering, graph partitioning, and sparse feature extraction. However, up to now a scalable implementation of PNMF for large-scale machine learning problems has been lacking. Here we provide an online algorithm for fast PNMF learning with low memory cost. The new algorithm simply applies multiplicative update rules iteratively on small subsets of the data, with historical data naturally accumulated. Consequently users do not need extra efforts to tune any optimization parameters such as learning rates or the history weight. In addition to scalability and convenience, empirical studies on synthetic and real-world datasets indicate that our online algorithm runs much faster than the existing batch version.

Keywords: Online learning, PNMF, NMF, large-scale datasets.

1 Introduction

Nonnegative Matrix Factorization (NMF) has attracted a lot of research attention since the initial work by Lee and Seung [6]. It is a method for efficiently and accurately generating low-rank approximations to large non-negative data matrices, which often occurs in practical applications. A multitude of NMF variants have been proposed later (see e.g. [2,3]). NMF and its variants have been applied to a variety of machine learning problems such as source separation [3], clustering [2], estimation in hidden Markov model [5], etc. See [1] for a survey.

To handle large-scale problems, a couple of scalable implementations of NMF have lately been introduced. Liu et al. presented a distributed NMF algorithm by carefully partitioning the data and arranging the computations to maximize data locality and parallelism [8]. Marial et al. proposed a framework for online matrix factorization based on the Euclidean distance, with NMF as its special case [9].

Recently, *Projective Nonnegative Matrix Factorization* (PNMF) as a new variant has achieved significant improvements over NMF. PNMF is able to produce a

* Supported by the Academy of Finland in the project Finnish Center of Excellence in Computational Inference Research (COIN).

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 285–290, 2012.

highly orthogonal or sparse factorizing matrix, which is desired in problems such as part-based feature extraction, clustering, and graph partitioning. Moreover, it has close relation to positive principal component analysis and K-means, with easy extension to nonlinear versions via kernels. The PNMF approximations of new data items, not included in the training set, can be computed efficiently without iterations. However, scaling up PNMF to large data matrices is more challenging because the approximate as a function is quadratic with respect to the factorizing matrix. Consequently the learning usually involves higher-order optimization problems. Previously there were only solutions to the cases where either dimension of the input matrix is small (see e.g. [12]).

In this paper we present an online algorithm for PNMF which is scalable to problems where both dimensions of input matrix are large. Given a small set of matrix columns, our algorithm applies a multiplicative update rule followed by a normalization step. Only one small matrix in addition to the factorizing matrix needs to be stored during the iterations. The accumulation of historical data is parameter-free in our approach. As a result, the new algorithm does not require extra efforts to tune any learning parameters, which facilitates its applications. The proposed method is not only scalable and convenient, but runs faster than the previously existing PNMF algorithm, as shown by experiments on both synthetic and real-world data.

2 Projective Nonnegative Matrix Factorization

Given a nonnegative input matrix $\mathbf{X} \in \mathbb{R}_+^{m\times n}$, whose columns are typically m-dimensional data vectors, one tries to find a nonnegative projection matrix $\mathbf{P} \in \mathbb{R}_+^{m\times m}$ of rank r such that

$$\mathbf{X} \approx \widehat{\mathbf{X}} \equiv \mathbf{P}\mathbf{X}. \tag{1}$$

In particular, *Projective Nonnegative Matrix Factorization* (PNMF) calculates the factorization

$$\mathbf{P} = \mathbf{W}\mathbf{W}^T, \tag{2}$$

where $\mathbf{W} \in \mathbb{R}_+^{m\times r}$. Compared with the Non-negative Matrix Factorization (NMF) [6] where $\mathbf{X} \approx \mathbf{W}\mathbf{H}$, PNMF replaces the second factorizing matrix $\mathbf{H}$ with $\mathbf{W}^T\mathbf{X}$. This brings PNMF close to non-negative Principal Component Analysis. A trivial solution $\mathbf{W} = \mathbf{I}$ appears when $r = m$, which will produce zero error but is practically useless. Useful PNMF results usually appear when $r \ll m$ for real-world applications.

The term “projective” refers to the fact that $\mathbf{W}\mathbf{W}^T$ would indeed be a projection matrix if $\mathbf{W}$ were an orthogonal matrix: $\mathbf{W}^T\mathbf{W} = \mathbf{I}$. It turns out that in PNMF learning, $\mathbf{W}$ becomes approximately, although not exactly, orthogonal. This has positive consequences in sparseness of the approximation, orthogonality of the factorizing matrix, decreased computational complexity in learning, close equivalence to clustering, generalization of the approximation to new data without heavy re-computations, and easy extension to a nonlinear kernel method with wide applications for optimization problems [12].

The approximation error in PNMF can be measured by the squared Euclidean distance:

$$D_{\mathrm{EU}}\left(\mathbf{X}||\mathbf{W}\mathbf{W}^T\mathbf{X}\right) = \sum_{ij}\left[X_{ij} - \left(\mathbf{W}\mathbf{W}^T\mathbf{X}\right)_{ij}\right]^2. \tag{3}$$

The original or batch PNMF algorithm that minimizes the above objective iteratively applies the update rule

$$W_{ik} \leftarrow W_{ik}\frac{\left(\mathbf{X}\mathbf{X}^T\mathbf{W}\right)_{ik}}{\left(\mathbf{W}\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W} + \mathbf{X}\mathbf{X}^T\mathbf{W}\mathbf{W}^T\mathbf{W}\right)_{ik}} \tag{4}$$

followed by the normalization

$$\mathbf{W} \leftarrow \mathbf{W}/\|\mathbf{W}\|, \tag{5}$$

where $\|\cdot\|$ takes the maximal singular value. Recently, Yang and Oja [13] gave a theoretically convergent batch algorithm for PNMF with a smaller exponent in the multiplicative update rule, which is however slower than the original algorithm in practice. Both the original and modified batch algorithms are not scalable to problems where both dimensions of input matrix are large.

3 Online Learning

Online optimization techniques such as stochastic gradient descent are commonly used for scaling-up machine learning algorithms. However, most of them are not suitable for PNMF because the objective function Eq. (3) is quartic with respect to $\mathbf{W}$. The additive updating methods require a parameter of learning step size, which is particularly difficult to choose in such a fourth-order optimization. Due to more costly gradients and Hessian, the automatic selection methods such as line search requires prohibative computation. Conventional principles for specifying learning rates such as the Robbins-Monroe rule [4] often lead to poor convergence speed. Though comprehensive parametric rules might work better, they require tedious work to tune the hyper-parameters.

Another drawback of additive updates is that they cannot guarantee the nonnegativity and an extra projection step is thus needed. Such projection can nevertheless be inconsistent with the gradient descent learning. In our practice, the *projected gradient* algorithms (e.g. [7]) that is popularly used in NMF behaves very unstably for PNMF and often has slow convergence.

Here we present an online multiplicative algorithm for PNMF. Though the PNMF gradient has a more comprehensive form, there is only one costly part which appears in all its terms:

$$\mathbf{Q} = \mathbf{X}\mathbf{X}^T\mathbf{W} \tag{6}$$

Algorithm 1. Online multiplicative algorithm for PNMF

```
Usage: $\mathbf{W} \leftarrow \mathrm{OnlinePNMF}(\mathbf{X}, r, p)$, where $p \ll n$.
Initialize $\mathbf{W} \in \mathbb{R}_+^{m \times r}$; calculate $\widetilde{\mathbf{Q}} = \mathbf{X}\mathbf{X}^T\mathbf{W}$.
repeat
    Form $\widetilde{\mathbf{X}}$ by sampling $p$ columns of $\mathbf{X}$.
    $\widetilde{\mathbf{Q}} \leftarrow \widetilde{\mathbf{Q}} + \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T\mathbf{W}$.
    $W_{ik} \leftarrow W_{ik} \frac{\widetilde{Q}_{ik}}{\left(\mathbf{W}\mathbf{W}^T\widetilde{\mathbf{Q}} + \widetilde{\mathbf{Q}}\mathbf{W}^T\mathbf{W}\right)_{ik}}$.
    $\mathbf{W} \leftarrow \mathbf{W}/\|\mathbf{W}\|$.
until stop conditions are satisfied
```

The expenses of other computations are negligible compared to it. We therefore apply online approximation of $\mathbf{Q}$ by taking a small subset of $\mathbf{X}$ columns, denoted by $\widetilde{\mathbf{X}}$, and replacing $\mathbf{Q}$ with $\widetilde{\mathbf{Q}} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T\mathbf{W}$ in each iteration. In this work, the subset is randomly sampled by uniform distribution. Following the merits of the multiplicative algorithm, we do not impose any learning step sizes in the update rule.

The above naive extension itself is not enough because $\widetilde{\mathbf{Q}}$ is calculated from scratch in each iteration. This totally discards the history in approximating $\mathbf{Q}$ and consequently the learning always proceeds with "cold starts" and zigzags a lot. Instead, we hope that the algorithm becomes more stable after certain warm-up stage where the learning history is accumulated. We implement this by a warm start technique [9] which adds the new quantity to the old one:

$$\widetilde{\mathbf{Q}} \leftarrow \widetilde{\mathbf{Q}} + \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T\mathbf{W}. \tag{7}$$

Though one may further apply some weighting between the old and new quantities, we have not observed significant improvement when using such weighting.

The resulting online PNMF algorithm is given in Algorithm 1. Compared with the old PNMF update rule in Eq. (4) which requires $O(m \times \min(m, n))$ memory to store $\mathbf{X}$ or $\mathbf{X}\mathbf{X}^T$, the new method only requires to store $\mathbf{W}$ and $\widetilde{\mathbf{Q}}$, both of size $O(m \times r)$. The computation time for each batch update is $O(mnr)$ or $O(m^2r)$, while for each online update is $O(mpr)$. Note that we can apply the same update rules for newly coming data to achieve incremental learning.

4 Experiments

Though online learning requires more iterations, as a whole it usually decreases the objective faster than batch updates. We verify this by comparing the online PNMF algorithm against the batch implementation on four datasets. The first two are generated from a Gaussian mixture with 16 components. The third and fourth consist of face images taken from the FERET [10] and the UND [11] databases. The dimensions of the input matrices are 1129×1000, 11235×10000, 1024×2409, and 10304×33247, respectively. For face images, PNMF produces a sparse feature basis.

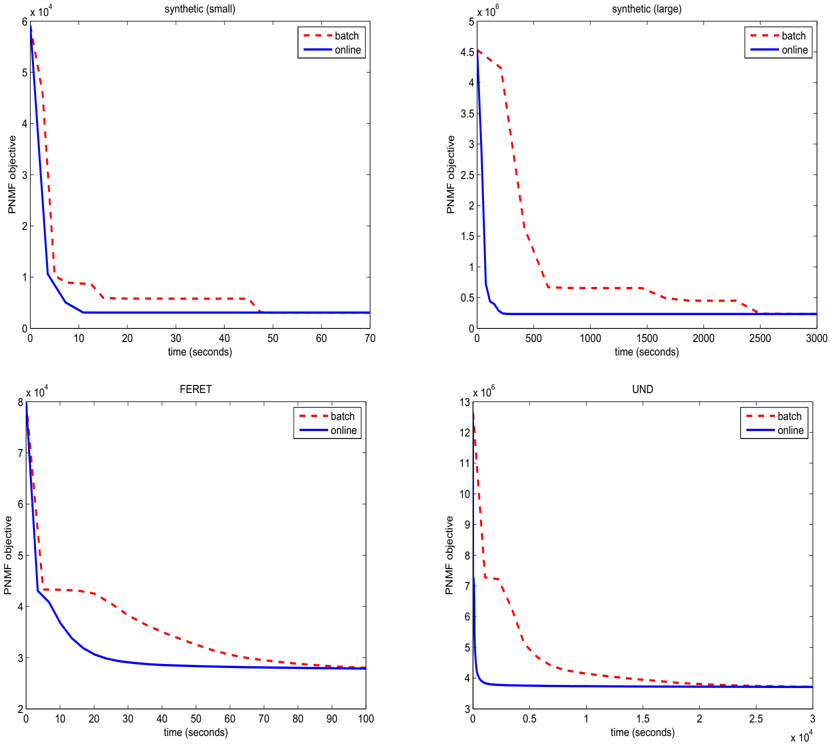


Fig. 1. Evolutions of PNMF objectives using the compared implementations

The objective evolution curves are shown in Figure 1. It can be clearly seen that the online algorithm defeats the batch version for all datasets. The speedup is even clearer in the two larger datasets *synthetic (large)* and *UND*, where the objectives using the new algorithm converge tens of times faster than those using the old implementation.

5 Conclusions

We have presented a convenient online algorithm for Projective Nonnegative Matrix Factorization. With low-memory cost, the proposed algorithm is scalable to large PNMF problems. The new method also demonstrated substantial advantage in fast PNMF training.

Subsampling methods could also be used to further reduce the computational cost in initialization. For offline training, one may also sparsely sample some entries of $\mathbf{X}$, not necessarily the columns, still with memory saving and speedup introduced by the stochastic gradients.

References

1. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis. John Wiley (2009)
2. Ding, C., Li, T., Jordan, M.I.: Convex and semi-nonnegative matrix factorizations. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(1), 45–55 (2010)
3. Févotte, C., Bertin, N., Durrieu, J.L.: Nonnegative matrix factorization with the Itakura-Saito divergence: with application to music analysis. Neural Computation 21(3), 793–830 (2009)
4. Kushner, H.J., Clark, D.S.: Stochastic Approximation Methods for Constrained and Unconstrained Systems. Springer, New York (1978)
5. Lakshminarayanan, B., Raich, R.: Non-negative matrix factorization for parameter estimation in hidden markov models. In: Proceedings of IEEE International Workshop on Machine Learning for Signal Processing, pp. 89–94 (2010)
6. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
7. Lin, C.J.: Projected gradient methods for non-negative matrix factorization. Neural Computation 19, 2756–2779 (2007)
8. Liu, C., Yang, H., Fan, J., He, L., Wang, Y.: Distributed nonnegative matrix factorization for web-scale dyadic data analysis on MapReduce. In: Proceedings of 19th International World Wide Web Conference (2010)
9. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. The Journal of Machine Learning Research 11, 19–60 (2010)
10. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face recognition algorithms. IEEE Trans. Pattern Analysis and Machine Intelligence 22, 1090–1104 (2000)
11. Woodard, D., Flynn, P.: Finger surface as a biometric identifier. Computer Vision and Image Understanding 100(3), 357–384 (2005)
12. Yang, Z., Oja, E.: Linear and nonlinear projective nonnegative matrix factorization. IEEE Transaction on Neural Networks 21(5), 734–749 (2010)
13. Yang, Z., Oja, E.: Unified development of multiplicative algorithms for linear and quadratic nonnegative matrix factorization. IEEE Transactions on Neural Networks 22(12), 1878–1891 (2011)

Optimization of Fuzzy Systems Using Group-Based Evolutionary Algorithm

Jyh-Yeong Chang*, Ming-Feng Han, and Chin-Teng Lin

Institute of Electrical Control Engineering
National Chiao Tung University
1001 University Road, Hsinchu, Taiwan 300, ROC
{jychang,ctlin}@mail.nctu.edu.tw, ming0901@gmail.com

Abstract. This paper proposes a group-based evolutionary algorithm (GEA) for the fuzzy system (FS) optimization. Initially, we adopt an entropy measure method to determine the number of rules. Fuzzy rules are automatically generated from training data by entropy measure. Subsequently, the GEA is performed to optimize all the free parameters for the FS design. In the evolution process, a FS is coded as an individual. All individuals based on their performance are partitioned into a superior group and an inferior group. The superior group, which is composed of individuals with better performance, uses a global evolution operation to search potential individuals. In the inferior group, individuals with a worse performance employ the local evolution operation to search better individuals near the current best individual. Finally, the proposed FS with GEA model (FS-GEA) is applied to time series forecasting problem. Results show that the proposed FS-GEA model obtains better performance than other algorithm.

Keywords: fuzzy system (FS), differential evolution (DE), group-based evolutionary algorithm (GEA), optimization.

1 Introduction

The fuzzy system (FS) [1-3] has become a popular research topic since its invention. Such system which has the high-level reasoning as the human thinking process is an efficient tool for solving complex problems. For the FS design, many papers have employed the backpropagation (BP) algorithm to train parameters. The BP algorithm is a powerful training technique that quickly minimises the error function of the FS. However, the BP algorithm may trap into a local minimum solution. To overcome this disadvantage, many researchers have applied the evolutionary algorithms (EA) to design a FS [4-10].

In EAs, the differential evolution (DE) algorithm has sparked the interest of researchers in recent years [11-17]. The DE algorithm, proposed by Storn and Price [11], is an efficient global optimizer in the continuous search domain. It has been shown to perform better than GA and PSO with respect to several numerical

* Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 291–298, 2012.

benchmarks [11-13]. However, the DE algorithm may favor the exploitation ability or the exploration ability [11]. An imbalance of evolution ability easily obtains lower performance for solving practical problems. To deal with this problem, previous studies have improved the mutation operation model. In [14-17], the researchers have proposed a modified differential evolution (MODE) algorithm for an adaptive neural fuzzy network design. This MODE algorithm provides a convex type mutation model and cluster-based scheme to increase the diversity of the population. In addition, the MODE algorithm has been applied to design the recurrent FS [17]. The concept of the tradeoff between the exploration ability and exploitation ability was proposed by Das *et al.*[11]. They designed a novel mutation model, called neighborhood-based mutation operation, to handle stagnation problem. In their paper, they utilized new mutation strategy and ring topology of neighborhood to find potential individuals in population. The neighborhood-based mutation operation also applied to the FS optimization [12]. However, a single evolution model may be limitative to deal with various problems [9,14].

This paper proposes a group-based evolutionary algorithm (GEA) to design the FS. The idea of the GEA is implemented on the DE algorithm. The proposed GEA employs the global evolution operation and the local evolution operation instead of single evolution model to effectively enhance the search ability. In the process of FS design, we adopt an entropy measure method to determine the number of rules for the NSF and identify suitable initial parameters. Subsequently, the GEA is performed to optimize all the free parameters for the FS design. In the simulation, the Mackey-Glass chaotic time series is conducted to evaluate the performance of the proposed FS with GEA (FS-GEA) model. Comparisons with other EAs demonstrate the superiority of the performance of the FS-GEA model.

This paper is organized as follows. Section 2 describes the basic structure and function for the FS. Section 3 introduces the rule generation and parameter optimization algorithm in the GEA. Section 4 presents the results of FS-GEA model and its performance comparisons with other paper. Finally, Section 5 draws conclusions.

2 The Architecture of the FS

The architecture of the FS is described in this section. The proposed FS realizes a nonlinear combination of input variables in consequent part. Each fuzzy rule corresponds to an output of functional link neural network (FLNN) [3,15,17], comprising a functional link. The proposed FS is five-layered network architecture which includes the input layer, membership function layer, rule layer, functional link layer and output layer. The operation functions of each layer are described as follows. In the following description, $O^{(p)}$ denotes the output of a node in the pth layer.

Layer 1—Input layer: No computation is done in this layer. Each node in this layer, which corresponds to one input variable, only transmits input values to the next layer directly. That is

$$O^{(1)} = x_i \qquad i=1,2,...,n \tag{1}$$

where n are the input variables of the FS.

Layer 2—Membership function layer: Each node in this layer is a membership function that corresponds one linguistic label of one of the input variables in Layer 1. In other words, the membership value which specifies the degree to which an input value belongs to a fuzzy set is calculated in Layer 2

$$O^{(2)} = \mu_{ij} = \exp\left({-(x_i - m_{ij})^2} \middle/ {\sigma_{ij}^2} \right) \tag{2}$$

where $j = 1, 2..., M$, M is number of rules in the FS, m_{ij} and σ_{ij} are the center and the width of the Gaussian membership function of input variable, respectively.

Layer 3—Rule layer: This layer receives 1-D membership degrees of the associated rule from the nodes of a set in layer 2. Here, the product operator described before is adopted to perform the precondition part of the fuzzy rules. As a result, the output function of each inference node is

$$O^{(3)} = R_j = \prod_{i=1}^{n} \mu_{ij} \quad . \tag{3}$$

The output of a layer 3 node represents the firing strength of the corresponding fuzzy rule.

Layer 4—Functional link layer: The input to a node in layer 4 is the output from layer 3, and the other inputs are calculated from a functional link neural network that has not used the function $tanh(\cdot)$. For such a node,

$$O^{(4)} = R_j (\sum_{k=1}^{l} w_{kj} \varphi_k) , \tag{4}$$

where w_{kj} is the corresponding link weight of functional link neural network and ϕ_k is the functional expansion of input variables. The functional expansion uses a trigonometric polynomial basis function, given by $[x_1 \; sin(\pi \, x_1) \; cos(\pi \, x_1) \, x_2 \; sin(\pi \, x_2) \, cos(\pi \, x_2)]$ for two-dimensional input variables. Therefore, l is the number of basis functions, $l = 3 \times n$, where n is the number of input variables. Moreover, the output nodes of functional link neural network depend on the number of fuzzy rules of the FS.

Layer 5—Output layer: Each node in this layer corresponds to one output variable. The node integrates all of the actions recommended by layer 3 and layer 4, which acts as a defuzzifier with

$$O^{(5)} = y = \frac{\sum_{j=1}^{M} R_j (\sum_{k=1}^{l} w_{kj} \varphi_k)}{\sum_{j=1}^{M} R_j} , \tag{5}$$

where M is the number of fuzzy rules, and y is the output of the FS.

3 Learning Process for the FS Design

This section describes structure learning and parameter learning for the FS design. A rule generation method based on entropy measure for structure learning is to automatically determine the number of clusters. After rule generation, all free parameters in the FS are learned by the GEA for parameter learning. Finally, a fuzzy system can be designed by our method.

3.1 Rule Generation Using Adaptive Entropy Measure Algorithm

In this paper, an adaptive entropy measure algorithm is proposed for rule generation. For each incoming pattern x_i, the rule firing strength can be regarded as the degree to which the incoming pattern belongs to the corresponding cluster. Entropy measure between each data point and each membership function is calculated based on a similarity measure. A data point of closed mean has lower entropy. Therefore, the entropy values between data points and current membership functions are calculated to determine whether or not to add a new rule. For computational efficiency, the entropy measure can be calculated using the firing strength from μ_{ij} as follows

$$EM_j = -\sum_{i=1}^{n} \exp(1/\mu_{ij}) \cdot \log_2(\exp(1/\mu_{ij})) , \tag{6}$$

where $EM_j \in [0,1]$. According to Eq. (6), the measure is used to generate a new fuzzy rule for new incoming data is described as follows. The maximum entropy measure

$$EM_{\max} = \max_{1 \le j \le M_{(t)}} EM_j \tag{7}$$

is determined, where $M_{(t)}$ is the number of existing rules at time t. If $EM_{\max} \le \overline{EM}$, then a new rule is generated, where $\overline{EM} \in [0,1]$ is a prespecified threshold that decays during the learning process. Once a new rule has been generated, the next step is to assign the initial mean and variance as follows;

$$m_{ij}^{(M_{(t+1)})} = x_i \tag{8}$$

$$\sigma_{ij}^{(M_{(t+1)})} = \sigma_{init} \tag{9}$$

where x_i is the new input and σ_{init} is a prespecified constant.

Until all entropy value satisfies a prespecified threshold, the process of rule generation is terminated. In this paper, the $\overline{EM}$ is defined as 0.26-0.3 times of the number of input variables [3].

3.2 The Learning Process of the GEA

For the effective parameter learning in the FS optimization, evolutionary algorithm is usually used [14-17]. In this paper, we propose a GEA to tune all free parameters for the FS optimization. The proposed GEA consists of six major steps：the coding step, population step, evaluation step, mutation step, crossover step, and selection step. The whole learning process is described as follows：

(1) Coding：The foremost step in the GEA algorithm is the coding of the FS into an individual. In this paper, an individual consists of the mean m_{ij} and width σ_{ij} of a Gaussian membership function, and w_{kj} weight of the consequent part, where *i* and *j* represent the *i*th input variable and the *j*th rule, respectively.

(2) Population：Before the GEA is performed, the individuals that will constitute an initial population must be created. The following formulations show the generation of the initial population.

$$\begin{aligned} FS_q &= [rule_1^q \mid rule_2^q \mid \ldots \mid rule_M^q] \\ &= [m_{i1}^* + \Delta m_{i1}^q, \sigma_{i1}^* + \Delta\sigma_{i1}^q, w_{k1}^q \mid \ldots \\ &\quad \mid m_{ij}^* + \Delta m_{ij}^q, \sigma_{ij}^* + \Delta\sigma_{ij}^q, w_{kj}^q \mid \ldots \\ &\quad \mid m_{iM}^* + \Delta m_{iM}^q, \sigma_{iM}^* + \Delta\sigma_{iM}^q, w_{kM}^q \ ] \end{aligned} \tag{10}$$

where m_{ij}^* and σ_{ij}^* are results of structure learning for the mean and width of the Gaussian membership function of the *j*th rule of the *i*th input variable, Δm_{ij}^q and $\Delta\sigma_{ij}^q$ are small random deviations that are uniformly generated from the interval [−0.1, 0.1], w_{kj} are randomly and uniformly generated from an interval whose range is identical to the FS output *y* range.

(3) Evaluation：In this paper, we adopt a fitness function to evaluate the performance of each individual. The fitness function used in this paper is the root mean-squared error (RMSE) between the desired and actual outputs.

(4) Mutation：Before the mutation operator, a sorting process arranges all individuals based on their fitness value as follows for minimum-objective problems: $fitness_1 < fitness_2 < \ldots < fitness_{NP-1} < fitness_{NP}$. According to fitness value, all individuals are partitioned into an inferior group and a superior group. The inferior group, with includes the NP/2 worst individuals, performs a global search to increase the diversity of the population and find a wide range of potential solutions. The other NP/2 individuals in the superior group perform a local search to actively detect better solutions near the current best solution. The following represents a complete mutation operation for the inferior group and the superior group.

$$\text{Inferior group}: \mathbf{V}_{i,gen} = \mathbf{X}_{i,gen} + F_i(\mathbf{X}_{r1,gen} - \mathbf{X}_{r2,gen}) \tag{11}$$

$$\text{Superior group}: \mathbf{V}_{i,gen} = \mathbf{X}_{gbest,gen} + F_i(\mathbf{X}_{r3,gen} - \mathbf{X}_{r4,gen}) \tag{12}$$

where F_i is scaling factors; $\mathbf{X}_{r1,gen}$, $\mathbf{X}_{r2,gen}$, $\mathbf{X}_{r3,gen}$ and $\mathbf{X}_{r4,gen}$ are randomly selected from the population; $i \neq r1 \neq r2 \neq r3 \neq r4$; and the $\mathbf{X}_{gbest,gen}$ is the best-so-far individual in the population. Next, the inferior group and the superior group are combined as a new population for the crossover operation and the selection operation.

(5) Crossover and Selection：After mutation operation, The GEA algorithm uses the crossover and selection operations to produce offsprings for the next generation. The crossover and selection operations follow the traditional operation in the DE algorithm [11].

4 Simulation

This section discusses a simulation which is considered to evaluate the proposed FS-GEA. The proposed FS-GEA model is applied to predict the Mackey-Glass chaotic time series. In this simulation, we set the population size *NP* =50, Initial *CR*=0.8, Initial *F*=0.5 and the number of generations = 2000. All results are obtained based on 20 independent runs. For a fair comparison, the DE, jDE and MODE are performed with the same parameters in this simulation.

The Mackey-Glass chaotic time series *x(t)* was generated using the following delay differential equation:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \cdot$$

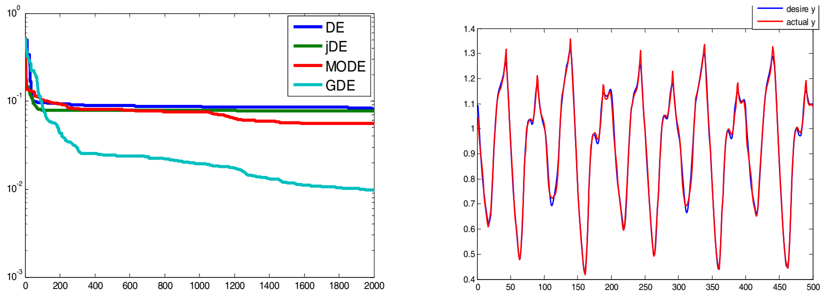
where τ > 17. As in previous studies [3], the parameter τ = 30, and x(0) = 1.2 in this simulation. Four past values are used to predict x(*t*), and the input–output pattern format is given by $[x(t-24), x(t-18), x(t-12), x(t-6) \mid x(t)]$.

Table 1. The best performance of the FS-GEA model and other methods

Method	Testing RMSE	Method	Testing RMSE
FS-GEA	**0.0075**	SEFC [23]	0.032
FLNFN-CCPSO[7]	0.0082	Back-propagation NN	0.02
RBF-AFS[18]	0.0128	Six-order polynomial	0.04
HyFIS[19]	0.01	Cascaded-correlation	0.06
NEFPROX[20]	0.053	Auto regressive model	0.19
D-FNN[21]	0.008	Linear predictive	0.55
GA-FLC [22]	0.26		

A total of 1000 patterns are generated from t = 124 to 1123, where the first 500 patterns [form $x(1)$ to $x(500)$] are used to train, and the last 500 patterns [form $x(501)$ to $x(1000)$] are used to test. After adaptive entropy measure algorithm, three fuzzy rules are generated. Fig. 1(a) shows the learning curves of the DE, jDE, MODE and GEA models. The learning curve of the DE and jDE models presented a stagnation situation after 150 generations. They trapped at local minimum solutions at training RMSE = 0.066 and 0.062. The MODE model slightly kept convergence results during evolution process. The proposed GEA model showed better learning curves than the other models. Fig.1(b) shows the prediction result of the proposed GEA model for the desired output and the actual output. The simulation result demonstrates the perfect predictive capability of the FS-GEA model.

A further comparison with other algorithms is shown in Table 1. Compared algorithms include Linear predictive, Auto regressive model, Cascaded-correlation, Six-order polynomial, Back-propagation NN, SEFC [23], GA-FLC [22], D-FNN[21], NEFPROX[20], HyFIS[19], RBF-AFS[18] and FLNFN-CCPSO[15]. The result predicted by the FS-GEA model is better than those predicting by other algorithms.



(a) (b)

Fig. 1. (a) learning curves of the DE, jDE, MODE and GEA models. (b)The prediction output of the FS-GEA model.

5 Conclusion

This study has proposed a GEA for the FS design. The adaptive entropy measure algorithm for rule generation helps to determine the number of rules and locate good initial parameters. All free parameters are learned by the GEA. The simulation result demonstrates that the FS-GEA model obtain a smaller RMSE than other evolutionary algorithms. Advanced topic on the proposed FS-GEA model should be addressed in the future research. The proposed FS-GEA model will be used to solve many practical problems, including brain signal based EEG prediction and cognitive state prediction problems in our laboratory.

Acknowledgements. This work was supported in part by the Aiming for the Top University Plan of National Chiao Tung University, the Ministry of Education, Taiwan, under Contract 100W9633 & 101W963.This work was also supported in part by the UST-UCSD International Center of Excellence in Advanced Bioengineering sponsored by the Taiwan National Science Council I-RiCE Program under Grant Number NSC-100-2911-I-009-101.

References

1. Lin, C.T., Lee, C.S.G.: Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent System. Prentice-Hall, Englewood Cliffs (1996)
2. Han, M.F., Lin, C.T., Chang, J.Y.: A Compensatory Neurofuzzy System with Online Constructing and Parameter Learning. In: Proc. of 2010 IEEE International Conference on Syst., Man, and Cybern., pp. 552–556 (2010)
3. Chen, C.H., Lin, C.J., Lin, C.T.: A Functional-Link-Based Neurofuzzy Network for Nonlinear System Control. IEEE Trans. Fuzzy Syst. 16, 1362–1378 (2008)
4. Juang, C.F., Chang, P.H.: Designing Fuzzy Rule-Based Systems Using Continuous Ant Colony Optimization. IEEE Trans. Fuzzy Syst. 18, 138–149 (2010)

5. Sanchez, E., Shibata, T., Zadeh, L.A.: Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives. World Scientific, Singapore (1997)
6. Chou, C.H.: Genetic Algorithm-Based Optimal Fuzzy Controller Design in The Linguistic Space. IEEE Trans. Fuzzy Syst. 14, 372–385 (2006)
7. Juang, C.F., Hsiao, C.M., Hsu, C.H.: Hierarchical Cluster-Based Multispecies Particle-Swarm Optimization for Fuzzy-System Optimization. IEEE Trans. Fuzzy Syst. 18, 14–26 (2010)
8. Juang, C.F.: A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design. IEEE Trans. Syst., Man, Cybern. B 34, 997–1006 (2004)
9. Li, C., Chiang, T.W.: Complex Fuzzy Model with PSO-RLSE Hybrid Learning Approach to Function Approximation. International Journal of Intelligent Information and Database Systems 5, 409–430 (2011)
10. Lin, C.J., Lee, C.Y.: Non-Linear System Control Using A Recurrent Fuzzy Neural Network Based on Improved Particle Swarm Optimization. International Journal of Systems Science 41, 381–395 (2010)
11. Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-Art. IEEE Trans. Evol. Comput. 14, 4–31 (2011)
12. Lin, C.T., Han, M.F., Lin, Y.Y., Liao, S.H., Chang, J.Y.: Neuro-Fuzzy System Design Using Differential Evolution with Local Information. In: 2011 IEEE International Conference on Fuzzy Systems, pp. 1003–1006 (2011)
13. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. IEEE Trans. Evol. Comput. 10, 646–659 (2006)
14. Han, M.F., Lin, C.T., Chang, J.Y.: Group-Based Differential Evolution for Numerical Optimization Problems. International Journal of Innovative Computing, Information and Control 9, 2 (2013)
15. Chen, C.H., Lin, C.J., Lin, C.T.: Nonlinear System Control Using Adaptive Neural Fuzzy Networks Based on a Modified Differential Evolution. IEEE Trans. Syst. Man Cybern. Part C, Appl. Rev., 459–473 (2009)
16. Lin, C.-J., Chen, C.-H., Lin, C.-T.: Efficient Self-Evolving Evolutionary Learning for Neurofuzzy Inference Systems. IEEE Trans. Fuzzy Syst. 16(6), 1476–1490 (2008)
17. Lin, C.-J., Wu, C.-F., Lee, C.-Y.: Design of a Recurrent Functional Neural Fuzzy Network Using Modified Differential Evolution. International Journal of Innovative Computing, Information and Control 7(1), 669–683 (2011)
18. Cho, K.B., Wang, B.H.: Radial Basis Function Based Adaptive Fuzzy Systems and Their Applications to System Identification. Fuzzy Sets Syst. 83, 325–339 (1996)
19. Kim, J., Kasabov, N.K.: HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamic systems. Neural Netw. 12, 1301–1319 (1999)
20. Nauk, D., Kruse, R.: Neuro-Fuzzy Systems for Function Approximation. Fuzzy Sets Syst. 101, 261–271 (1999)
21. Wu, S., Er, M.J.: Dynamic Fuzzy Neural Networks—A Novel Approach to Function Approximation. IEEE Trans. Syst., Man, Cybern. B 30, 358–364 (2000)
22. Karr, C.L.: Design of An Adaptive Fuzzy Logic Controller Using A Genetic Algorithm. In: Proc. 4th Conf. Genetic Algorithms, pp. 450–457 (1991)
23. Juang, C.F., Lin, J.Y., Lin, C.T.: Genetic Reinforcement Learning Through Symbiotic Evolution for Fuzzy Controller Design. IEEE Trans. Syst., Man, Cybern., Part B 30, 290–302 (2000)

Gabor-Based Novel Local, Shape and Color Features for Image Classification

Atreyee Sinha*, Sugata Banerji, and Chengjun Liu

Department of Computer Science,
New Jersey Institute of Technology,
Newark, NJ 07102, USA
{as739,sb256,chengjun.liu}@njit.edu
http://cs.njit.edu/liu

Abstract. This paper introduces several novel Gabor-based local, shape and color features for image classification. First, a new Gabor-HOG (GHOG) descriptor is proposed for image feature extraction by concatenating the Histograms of Oriented Gradients (HOG) of all the local Gabor filtered images. The GHOG descriptor is then further assessed in six different color spaces to measure classification performance. Finally, a novel Fused Color GHOG (FC-GHOG) feature is presented by integrating the PCA features of the six color GHOG descriptors that performs well on different object and scene image categories. The Enhanced Fisher Model (EFM) is applied for discriminatory feature extraction and the nearest neighbor classification rule is used for image classification. The robustness of the proposed GHOG and FC-GHOG feature vectors is evaluated using two grand challenge datasets, namely the Caltech 256 dataset and the MIT Scene dataset.

Keywords: The Gabor-HOG (GHOG) descriptor, Fused Color GHOG (FC-GHOG) descriptor, Histograms of Oriented Gradients (HOG), Gabor filters, Principal Component Analysis (PCA), Enhanced Fisher Model (EFM), Color spaces, Image search.

1 Introduction

Color contains more discriminating information than grayscale images [1], and color based image search can be very effective for image classification tasks [2], [3], [4]. Some desirable properties of descriptors defined in different color spaces include relative stability over changes in photographic conditions such as varying illumination. Global color features such as the color histogram and local invariant features provide varying degrees of success against image variations such as rotation, viewpoint and lighting changes, clutter and occlusions [5], [6]. Shape and local features also provide important cues for content based image classification and retrieval. Local object appearance and shape within an image can be

* Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 299–306, 2012.

described by the Histograms of Oriented Gradients (HOG) that stores distribution of edge orientations within an image [7]. Several researchers have described the biological relevance and computational properties of Gabor wavelets for image analysis [8], [9]. Lades et al. [10] used Gabor wavelets for face recognition using the Dynamic Link Architecture (DLA) framework. Lately, Donato et al. [11] showed experimentally that the Gabor wavelet representation is optimal for classifying facial actions.

The motivation behind this work lies in the concept of how people understand and recognize images. We subject the image to a series of Gabor wavelet transformations, whose kernels are similar to the 2D receptive field profiles of the mammalian cortical simple cells [8]. The novelty of this paper is in the construction of several feature vectors based on Gabor filters. Specifically, we first present a novel Gabor-HOG (GHOG) descriptor by concatenating the Histograms of Oriented Gradients (HOG) of the components of the images produced by the result of applying Gabor filters in different scales and orientations. We then assess our GHOG feature vector in six different color spaces and propose several new color GHOG feature representations. We further extend this concept by integrating the six color GHOG features using a fusion technique that implements feature extraction by means of PCA to produce the novel Fused Color GHOG (FC-GHOG) descriptor. Discriminatory feature extraction applies the Enhanced Fisher Model (EFM) [12], and image classification is based on the nearest neighbor classification rule. Finally, the effectiveness of the proposed descriptors and classification method is evaluated using two datasets: the Caltech 256 grand challenge image dataset and the MIT Scene dataset.

2 Gabor-Based Novel Local, Shape and Color Features for Image Classification

This section discusses the proposed novel descriptors and classification methodology for image classification.

2.1 The Gabor-HOG (GHOG) and Fused Color GHOG (FC-GHOG) Descriptors

A Gabor filter is obtained by modulating a sinusoid with a Gaussian distribution. In a 2D scenario such as images, a Gabor filter is defined as:

$$g_{\nu,\theta,\phi,\sigma,\gamma}(x',y') = \exp(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2})\exp(i(2\pi\nu x'+\phi)) \qquad (1)$$

where $x' = x\cos\theta + y\sin\theta$, $y' = -x\sin\theta + y\cos\theta$, and ν, θ, ϕ, σ, γ denote the spatial frequency of the sinusoidal factor, orientation of the normal to the parallel stripes of a Gabor function, phase offset, standard deviation of the Gaussian kernel and the spatial aspect ratio specifying the ellipticity of the support of the Gabor function respectively.

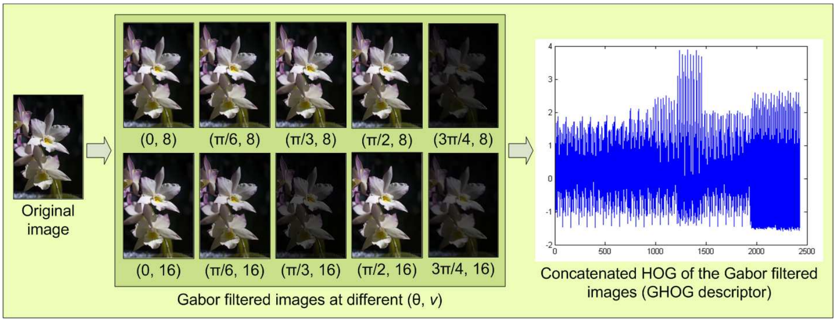


Fig. 1. The generation of the proposed GHOG descriptor

For a grayscale image $f(x, y)$, the Gabor filtered image is produced by convolving the input image with the real and imaginary components of a Gabor filter. Considering that the Gabor wavelet representation captures the local structure corresponding to spatial frequency (scale), spatial localization, and orientation selectivity [13], [14] we used multi-resolution and multi-orientation Gabor filtering for subsequent extraction of feature vectors. We subject each of the three color components of the image to ten combinations of Gabor filters with two scales (spatial frequencies) and five orientations. For our experiments, we choose $\phi = 0$, $\sigma = 2$, $\gamma = 0.5$, $\theta = [0, \pi/6, \pi/3, \pi/2, 3\pi/4]$, and $\nu = [8, 16]$.

We concatenate the HOG of the color components of the resultant filtered images and normalize to zero mean and unit standard deviation to produce a new Gabor-HOG (GHOG) image descriptor. Figure 1 illustrates the creation of the GHOG feature, the performance of which is measured on six different color spaces, namely RGB, HSV, oRGB [15], YCbCr, YIQ and DCS [16] as well as on grayscale. Figure 2 shows the grayscale and the color components of a sample

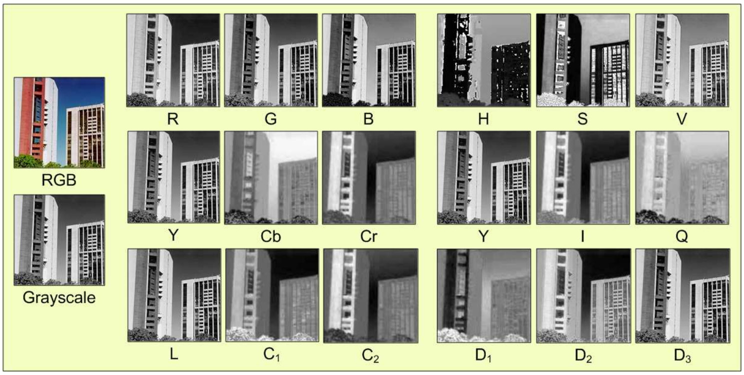


Fig. 2. A sample image from the MIT Scene dataset (labeled RGB) is shown split up into various color components of the RGB, HSV, YCbCr, YIQ, oRGB and DCS

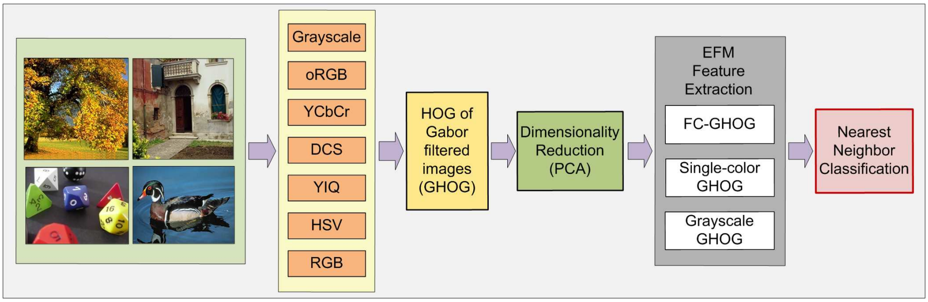

Grayscale
oRGB
YCbCr
DCS
YIQ
HSV
RGB
HOG of Gabor filtered images (GHOG)
Dimensionality Reduction (PCA)
EFM Feature Extraction
FC-GHOG
Single-color GHOG
Grayscale GHOG
Nearest Neighbor Classification

Fig. 3. An overview of multiple features fusion methodology, the EFM feature extraction method, and the classification stages

image in the six color spaces used by us in this paper. For fusion, we first use PCA for the optimal representation of our color GHOG vectors with respect to minimum mean square error. We then integrate the PCA features of the six normalized color GHOG descriptors to form the novel Fused Color GHOG (FC-GHOG) descriptor which outperforms the classification results of the individual color GHOG features.

2.2 The EFM-NN Classifier

We perform learning using Enhanced Fisher Linear Discriminant Model (EFM) [12] and classification is implemented using the nearest neighbor rule. The EFM method first applies Principal Component Analysis (PCA) to reduce the dimensionality of the input pattern vector. A popular classification method that achieves high separability among the different pattern classes is the Fisher Linear Discriminant (FLD) method. The FLD method, if implemented in an inappropriate PCA space, may lead to overfitting. The EFM method, which applies an eigenvalue spectrum analysis criterion to choose the number of principal components to avoid overfitting, improves the generalization performance of the FLD. The EFM method thus derives an appropriate low dimensional representation from the GHOG descriptor and further extracts the EFM features for pattern classification. We compute similarity score between a training feature vector and a test feature vector using the cosine similarity measure and classification

Table 1. Comparison of the classification performance (%) with other methods on Caltech 256 dataset. Note that [17] used 250 of the 256 classes with 30 training samples per class.

#train	**#test**	**GHOG**		**[4]**		**[17]**	
12800	6400	YCbCr	**30.2**	oRGB-SIFT	23.9		
		YIQ	**30.7**	CSF	30.1		
		FC	33.6	CGSF	**35.6**	SPM-MSVM	34.1

Fig. 4. Some sample images from the Caltech 256 dataset

is performed using the nearest neighbor rule. Figure 3 gives an overview of multiple feature fusion methodology, the EFM feature extraction method, and the classification stages.

3 Experimental Results

3.1 Caltech 256 Dataset

The Caltech 256 dataset [17] holds 30,607 images divided into 256 object categories and a clutter class. The images have high intra-class variability and high object location variability. Each category contains at least 80, and at most 827 images. The mean number of images per category is 119. The images represent a diverse set of lighting conditions, poses, backgrounds, and sizes. Images are in color, in JPEG format with only a small percentage in grayscale. The average

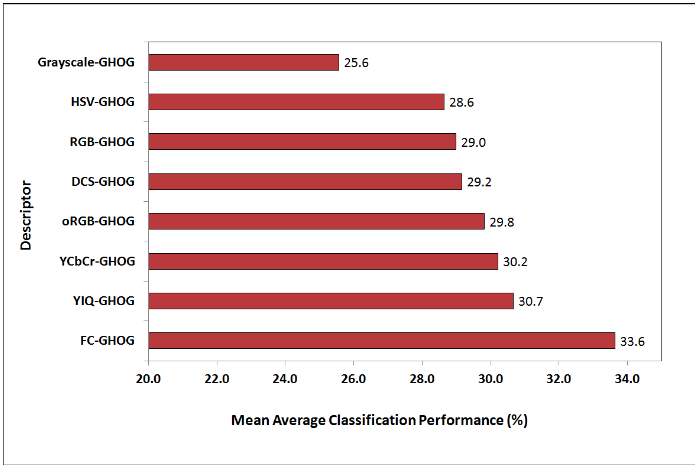


Fig. 5. The mean average classification performance of the proposed GHOG descriptor in individual color spaces as well as after fusing them on the Caltech 256 dataset

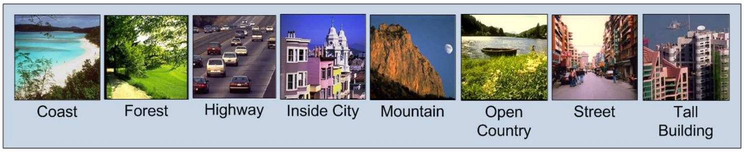


Fig. 6. Some sample images from the MIT Scene dataset

size of each image is 351 × 351 pixels. Figure 4 shows some sample images from this dataset.

For each class, we use 50 images for training and 25 images for testing. The data splits are the ones that are provided on the Caltech website [17]. In this dataset, YIQ-GHOG performs the best among single-color descriptors giving 30.7% success followed by YCbCr-GHOG with 30.2% classification rate. Figure 5 shows the success rates of the GHOG descriptors for this dataset. The FC-GHOG descriptor here achieves a success rate of 33.6%. Table 1 compares our results with those of SIFT-based methods.

3.2 MIT Scene Dataset

The MIT Scene dataset [18] has 2,688 images classified as eight categories: coast, forest, mountain, open country, highway, inside of cities, tall buildings, and streets. See figure 6 for some sample images from this dataset. All of the images are in color, in JPEG format, and of size 256 × 256 pixels. There is a large variation in light and angles along with a high intra-class variation.

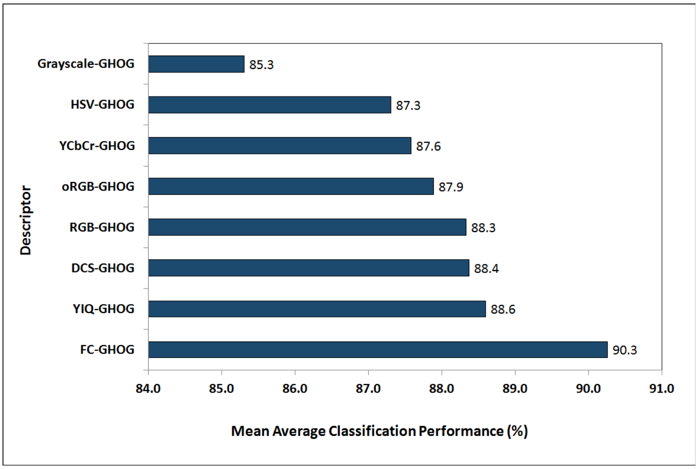


Fig. 7. The mean average classification performance of the proposed GHOG descriptor in individual color spaces as well as after fusing them on the MIT Scene dataset

Table 2. Category wise descriptor (GHOG) performance (%) on the MIT Scene dataset. Note that the categories are sorted on the FC-GHOG results.

Category	**FC**	**YIQ**	**DCS**	**RGB**	**oRGB**	**YCbCr**	**HSV**	**Grayscale**
forest	**98**	**98**	96	97	96	96	97	97
coast	**94**	91	88	90	90	90	88	87
inside city	91	92	**93**	91	**93**	92	90	91
street	90	89	**91**	90	88	88	84	88
tall building	**90**	89	86	87	88	88	87	84
mountain	**90**	86	86	88	87	87	85	79
highway	**88**	86	**88**	**88**	86	82	**88**	84
open country	**81**	77	78	76	77	78	79	73
Mean	**90.3**	**88.6**	**88.4**	**88.3**	**87.9**	**87.6**	**87.3**	**85.3**

Table 3. Comparison of the classification performance (%) with other methods on the MIT Scene dataset

#train	**#test**	**GHOG**		**[2]**		**[18]**
2000	688	DCS	**88.4**	CLF	86.4	-
		YIQ	**88.6**	CGLF	86.6	
		FC	**90.3**	CGLF+PHOG	89.5	
800	1888	YIQ	**84.7**	CLF	79.3	
		RGB	**84.9**	CGLF	80.0	
		FC	**86.9**	CGLF+PHOG	84.3	83.7

From each class, we use 250 images for training and the rest of the images for testing the performance, and we do a five-fold cross validation. Here too, YIQ-GHOG is the best single-color descriptor at 88.6%. DCS-GHOG also performs well to achieve 88.4% success rate. The combined descriptor FC-GHOG gives a mean average performance of 90.3%. See Figure 7 for details. Table 3 compares our result with that of other methods. Table 2 shows the class wise classification rates for the proposed GHOG descriptors on this dataset.

4 Conclusion

We have presented new Gabor-based local, shape and color feature extraction methods inspired by HOG for color images which exceed or achieve comparable performance to some of the best classification performances reported in the literature. Experimental results carried out using two grand challenge datasets show that the fusion of multiple color GHOG descriptors (FC-GHOG) achieves significant increase in the classification performance over individual color GHOG descriptors, which indicates that various color GHOG descriptors are not fully redundant for image classification tasks.

References

1. Gonzalez, R., Woods, R.: Digital Image Processing. Prentice Hall (2001)
2. Banerji, S., Verma, A., Liu, C.: Novel Color LBP Descriptors for Scene and Image Texture Classification. In: 15th International Conference on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas, Nevada (2011)
3. Shih, P., Liu, C.: Comparative Assessment of Content-based Face Image Retrieval in Different Color Spaces. International Journal of Pattern Recognition and Artificial Intelligence 19(7) (2005)
4. Verma, A., Banerji, S., Liu, C.: A New Color SIFT Descriptor and Methods for Image Category Classification. In: International Congress on Computer Applications and Computational Science, Singapore, pp. 819–822 (2010)
5. Burghouts, G., Geusebroek, J.M.: Performance Evaluation of Local Color Invariants. Computer Vision and Image Understanding 113, 48–62 (2009)
6. Stokman, H., Gevers, T.: Selection and Fusion of Color Models for Image Feature Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(3), 371–381 (2007)
7. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), Washington, DC, USA, vol. 1, pp. 886–893 (2005)
8. Marcelja, S.: Mathematical Description of the Responses of Simple Cortical Cells. Journal of the Optical Society of America 70, 1297–1300 (1980)
9. Daugman, J.: Two-Dimensional Spectral Analysis of Cortical Receptive Field Profiles. Vision Research 20, 847–856 (1980)
10. Lades, M., Vorbruggen, J., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R.P., Konen, W.: Distortion Invariant Object Recognition in the Dynamic Link Architecture. IEEE Transactions on Computers 42, 300–311 (1993)
11. Donato, G., Bartlett, M., Hager, J., Ekman, P., Sejnowski, T.: Classifying Facial Actions. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(10), 974–989 (1999)
12. Liu, C., Wechsler, H.: Robust Coding Schemes for Indexing and Retrieval from Large Face Databases. IEEE Transactions on Image Processing 9(1), 132–137 (2000)
13. Schiele, B., Crowley, J.: Recognition Without Correspondence Using Multidimensional Receptive Field Histograms. International Journal of Computer Vision 36(1), 31–50 (2000)
14. Liu, C., Wechsler, H.: Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition. IEEE Transactions on Image Processing 11(4), 467–476 (2002)
15. Bratkova, M., Boulos, S., Shirley, P.: oRGB: A Practical Opponent Color Space for Computer Graphics. IEEE Computer Graphics and Applications 29(1), 42–55 (2009)
16. Liu, C.: Learning the Uncorrelated, Independent, and Discriminating Color Spaces for Face Recognition. IEEE Transactions on Information Forensics and Security 3(2), 213–222 (2008)
17. Griffin, G., Holub, A., Perona, P.: Caltech-256 Object Category Dataset. Technical Report 7694, California Institute of Technology (2007)
18. Oliva, A., Torralba, A.: Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. International Journal of Computer Vision 42(3), 145–175 (2001)

A Hybrid KNN-Ant Colony Optimization Algorithm for Prototype Selection

Amal Miloud-Aouidate and Ahmed Riadh Baba-Ali

Laboratory of Robotics, Parallelism and Embedded
University of Sciences and Technology Houari Boumediene, USTHB,
BP 32 EL ALIA, BAB EZZOUAR
Algiers, Algeria
miloudamal@gmail.com, riadhbabaali@yahoo.fr

Abstract. The condensing KNN is the application of the K-Nearest Neighbors classifier with a condensed training set, which is a consistent subset calculated from the initial training set. In this work we present a novel algorithm, Ant-KNN, which allows improving the performance of the standard KNN classifier by a method based on ant colonies optimization. The results obtained through tests conducted on five benchmarks from UCI Machine Learning Repository demonstrate the improvement obtained by our algorithm in comparison with other condensing KNN algorithms.

Keywords: KNN, Ant colonies, Condensing, Prototype reduction, Prototype selection, Noise filtering.

1 Introduction

The K-nearest neighbor classification rule (KNN) is a powerful classification method allowing the classification of an unknown prototype using a set of training prototypes. The calculation of a consistent training subset with a minimal cardinality for the KNN rule [3] turns out to be hard [7]. Researchers have proposed the Prototypes Selection to address this problem.

One of the proposed selection methods was the "condensing" [6]. A condensing algorithm tries to determine a significantly reduced set of prototypes such that the classification accuracy of the 1-NN rule using this set as a training set must be close to the one reached on the complete training set. In this paper we will only consider condensing approaches. Below is a brief overview of some existing algorithms for the condensing approaches [13].

Hart in 1968 [14] was the first to propose a method for reducing the size of stored data for the nearest neighbor decision rule. The novelty of this method, "The Condensed Nearest Neighbor Rule" (CNN, compared to the conventional KNN) is the process of reducing the initial training set. The rule minimizes the number of prototypes by eliminating very similar training prototypes, and those that do not add additional information for classification.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 307–314, 2012.

The “Reduced Nearest Neighbor rule” (RNN) introduced by Gates [8], is an extension of the CNN rule. The RNN algorithm considers the consistency of classification in the original set rather than in the final set as proposed in the CNN.

Angiulli introduced the “Fast Condensed Nearest Neighbor rule” (FCNN) [1] which is a scalable algorithm on large multidimensional data sets, used to create subsets serving as consistent training sets based on the nearest neighbor decision rule. This algorithm allows selecting points very close to the decision boundaries (border between two classes), it is independent of the order, and has low quadratic complexity.

Wilson and Martinez suggested [15] a set of six algorithms for sets reduction based on the KNN algorithm. The first reduction technique presented was the DROP1 which represents an improvement of the RNN rule, and verifies the accuracy of the resulting set S instead of the initial set T. DROP2 sorts S in an attempt to remove the central points before the border points (points which are near from the decision boundaries). DROP3 uses a noise filtering before sorting the prototypes of S. DROP4 improves DROP3 rule and provides that a prototype is removed only if it is misclassified by its k nearest neighbors, and its removal does not affect the classification of other prototypes DROP5 upgrades DROP2 by proposing that the prototypes are considered beginning from the ones closest to the nearest enemy (an enemy is the nearest neighbor of a prototype from a different class) and proceeding to outside. The latest algorithm proposed by Wilson and Martinez [15] was the DEL which is similar to DROP3, except that it uses the length coding heuristic for deciding whether a prototype can be removed or not.

Wu, Ianakiev and Govindraju proposed the “Improved K-Nearest Neighbor Classification” [16] solution to increase the speed of traditional KNN classification while maintaining its level of accuracy by proposing two building techniques. The suggested IKNN algorithm is based on iterative elimination of prototypes with high attraction capacity.

Fayed and Atia [6] proposed the TRKNN which is a way to alleviate the reduction problem through a condensation approach. The aim of their approach was to eliminate the reasons that makes load the calculation and does not contribute to improve the classification.

Wu, Nikolaidis and Goulermas [12] introduced a new approach, "The Class Boundary Preserving Algorithm" (CBP), a multi-step method for pruning the training set. The proposed method aims to preserve prototypes that are close to the borders of classes.

In this paper, we introduce a new condensing algorithm called Ant-KNN. The basic idea is to define a shorter chain between the elements of a class using an Ant colonies optimization method, and eliminate, from this chain, the elements that do not add additional information to the classification. The rest of the paper is organized as follows. In Section 2, the proposed method is described and its main properties are stated. In Section 3, experimental results are presented together with a thorough comparison with existing methods. In Section 4, conclusions and future works are drawn.

2 Our Contribution

In this work we aimed to keep the minimum training elements for a correct classification, by eliminating elements that do not add information to the correct classification of the other elements of the initial training set. For this we proposed a three steps algorithm: the creation of the associate's chain, the application of the condensing algorithm and the application of the noise filter algorithm. We have used the ant colonies optimization [1] [3] [4] [6] to create the associate's chain connecting the prototypes through the shortest route possible.

2.1 Associate's Chain

An associate's chain Ci of a class wi is defined as the sequence xi0, xi1, xi2, …, xik-1 where k=|wi|, the element xi0 is the center of a class wi, and xij is the element of the same class reached by the shortest route from xij-1.

The sequence d_{i0}, d_{i1}, d_{i2}, …, d_{ik-2}, where the Euclidean distance between two elements x_{ij+1} and x_{ij} defined by d_{ij}=||x_{ij}+ 1-x_{ij}|| is the one used by the ACO algorithm to define the path linking the elements of a class w_i. The creation of the chain is stopped when the next element is the root element of this chain.

Conventions
X: set of elements;
n = | X |: number of elements;
b_i (t): number of ants in the element i at the instant t;
$n = \sum_{x \in i} b_i$: total number of ants;
n_{ij}=$\frac{1}{d_{ij}}$: visibility of an element j to an ant when it is in element i;
τ_{ij} (t): value of pheromone on the arc (i, j);

At any instant t, each ant chooses a destination element according to a defined choice probability. All ants are placed at the moment t +1 in an element of their choice. An iteration of the algorithm is defined by all the movements of the entire colony between t and t +1. Thus, after n iterations, the entire colony has done a Hamiltonian circuit [10] [11].

Transitions Choice
The ant k positioned on the element i will choose its destination j depending on the rate of visibility n_{ij} and the pheromone τ_{ij}. This choice is made with a probability of choosing the element j:

$$\mathrm{Prob}^{k}{}_{ij}(t)=\begin{cases} \frac{\tau ij(t)*nij}{\sum_{l \in N_i^k (t)} \tau_{il}*n_{il}} & \text{if j } N_i^k \\ 0 & else \end{cases} \quad (1)$$

where N_i^k is the set of cities that the ant k positioned on the element i still has not visited at the instant t.

Deposit pheromone

An ant deposits an amount of pheromone on each edge $\tau^{k}{}_{ij}$ of its route:

$$\tau^{k}{}_{ij} = \begin{cases} \Delta\tau_{ij}^{k}(t) = \frac{Q}{L^{k}(t)} & if\ (i,j) \in T^{k}(t) \\ o\ \ else \end{cases} \qquad (2)$$

Where $T^k(t)$ is the tour done by the ant k at the iteration t, $L^k(t)$ is the path length and Q a setting parameter.

Update pheromone

At the end of each iteration of the algorithm, pheromones deposited by ants in previous iterations evaporate following a hint: $\rho * \tau_{ij}(t)$, where $\rho \in [0,1]$.

At the end of the iteration the sum of pheromones that have not evaporated and those which have been deposited is calculated:

$$\tau_{ij}(t+1) = (1-\rho)\ \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \qquad (3)$$

The main steps used to create an associate's chain are:

For each class of the training set: create a chain linking all the elements along the shortest path:

1. For each element, calculate the distances to all other elements
2. Initialize the tour by designating the root (the center of the class)
3. Create as many ants as elements
4. Each ant chooses its destination element according to formula (1).
5. Each ant calculates the value $L^k(t)$.
6. The values $\tau^{k}{}_{ij}(t)$ are calculated according to formula (2).
7. The values of pheromone $\tau_{ij}(t)$ are updated according to the formula (3). In other words, the ant turns again in the opposite direction while depositing pheromones.
8. We look for the better tour than the best tour until now, and we store it (ie we look for an ant k such that $k = \min^{m}{}_{k=1} L^{k}(t)$).
9. The memories of ants (list of visited elements) are cleared.

2.2 Condensing Algorithm

The basic idea of the proposed condensing approach is as follows: For each class w_i in the training set, we construct the corresponding associate's chain C_i. An element x_{ij} of a chain is ignored if it does not add additional information to the classification of the elements of the initial training set.

For this, the k neighbors of each element of a chain are evaluated starting with those of the root element x_{i0}.

The final set F_i is created incrementally. It is initialized to x_{i0}. The set of covered neighbors (k neighbors of the elements of F_i) N_i will contain a single occurrence of neighbors covered by the elements of the final condensed set F_i. It is initialized to the k neighbors of the element x_{i0}. The chain is scanned element by element, and an element x_{ij} is added to F_i if it covers at least one non-existent neighbor in N_i, if so, non-existent neighbors in N_i are added to it. Otherwise the element is ignored.

We do so until the end of the chain C_i.

The main steps of this algorithm are:

1. For each class w_i we construct an associate's chain C_i.
2. For each element x_i C_i we evaluate its k nearest neighbors
3. Then we create F_i the reduced C_i
4. Finally we create F the final condensed set by combining all the reduced associate's chains of all classes.

Stopping Criterion

The algorithm is iterated until one of two criteria is met:

The class size falls below a threshold = k (the number of neighbors evaluated when condensing).

Or the accuracy reaches a threshold = 80%

2.3 Noise Filter

The last step of the proposed approach is the noise filtering. After the condensing step the resulting set is filtered. Prototypes that cause misclassification of the prototypes of the original training set are removed.

All the prototypes in the original training set O are re-classified using the new condensed set F as the 1-NN training set. When a prototype misclassifies another prototype (or more) this prototype is removed temporary. A reclassification of the prototypes in O is done using the condensed set but this prototype. If the classification rate reached using the filtered set is greater or equal to the classification rate reached using the condensed set before filtering, the prototype is definitely removed, and it is maintained otherwise.

The main steps of the noise filtering algorithm can be described as follows:

1. Classify prototypes in O using 1-NN and F
2. Calculate class_rate
3. For all prototypes j that misclassify a prototype in O do
4. re-classify O using 1-NN and F-{j}
5. calculate class_rate_N
6. If class_rate_N is greater or equal to class_rate then F=F-{j}

3 Experimental Results

The performances of the proposed condensing approach were tested on five different benchmarks downloaded from the UCI machine learning data repository. These datasets are: Ecoli, Glass, Heart (Cleveland), Heberman's Survival, and Iris. Table 1 presents the details of these seven datasets.

Table 1. Details of the 5 datasets used in the experiments, including the total number of prototypes, the dimensionality and the number of classes

Dataset	**Number of prototypes**	**Dimensional ity**	**Number of classes**
Ecoli	336	9	8
Glass	214	11	6
Heart (C)	303	14	5
Heberman	306	4	2
Iris	150	4	3

3.1 Description of the Experimental Plan

The experimental plan was designed to help verify the accuracy and the efficiency of the proposed method.

The plan stipulates that the training set is divided into ten equal parts. Nine parts together form the training set to reduce and the tenth part is used for validation.

To create the associate's chain the number of neighbors evaluated has been defined experimentally for Ecoli, Glass, Heart (Cleveland) and Iris to k = 16, and for Heberman's survival to k = 9. We have used a 1-NN classifier for classification.

3.2 Results

To validate our work, we compared the obtained test results to those found in the literature, from the same procedure as ours of the standard KNN and two algorithms presented in Section 1 that are TRKNN, and DROP3. The condensing rate (Cond) and the classification accuracy (Class) of each algorithm are reported in Table 2.

Table 2. Average accuracy and condensation percentages of the proposed Ant-KNN and two other compared algorithms over the five datasets

Dataset	**KNN**		**Ant-KNN**		**TRKNN**		**DROP3**	
	Cond	**Class**	**Cond**	**Class**	**Cond**	**Class**	**Cond**	**Class**
Ecoli	100	**100.00**	**07.59**	81.18	44.26	72.66	07.84	79.55
Glass	100	73.83	**10.41**	**100**	60.72	81.42	23.88	65.02
Heart (C)	100	81.19	**2.19**	**93.33**	73.54	58.67	12.76	80.84
Heberman	100	86.66	**11.95**	**93.33**	55.62	64.31	19.10	66.60
Iris	100	94.00	18.36	**100**	54.79	93.33	**14.81**	95.33
Mean	**100**	**87.13**	**10.10**	**93.56**	**57.78**	**74.07**	**15,67**	**77,46**

Table 2 shows that Ant-KNN presents the best accuracy and condensing means.

For the five benchmarks Ant-KNN presents the best classification rates (81.18%, 100%, 93.33%, 93.33% and 100%) compared to TRKNN and DROP3.

Compared to the standard KNN, the Ant-KNN presents the best accuracy for four datasets: Glass, Heart (C), Heberman and Iris.

The standard KNN presents an accuracy rate greater of 18.82% then the rate presented by the Ant-KNN for Ecoli.

The Ant-KNN presents the best condensing rates for four datasets: Ecoli, Glass, Heart (C) and Heberman compared to the two other condensing algorithms.

The Ant KNN follows DROP3 for Iris's condensing rate by an average difference of 3.55%, which represents a very low difference.

3.3 Analysis of Results

The algorithm was iterated between one and four times for the various datasets in order to improve the results.

Classification Analysis. The results showed that the algorithm presents high rates of classification accuracy averaging the 93.56% exceeding the average rates of the algorithms used for comparison.

Condensing Analysis. The proposed algorithm provides a very high reduction capacity averaging 10.10%. This is due to the creation of the associate's chain that can eliminate only the non-active elements for classification.

The negative point of this algorithm is its high complexity, which is an obstacle to its use on large datasets. This method is designed for applications in noisy data classification and for medium or small datasets classification.

4 Conclusions and Future Work

In this paper we presented a new approach for condensing kNN eliminating the elements that do not provide additional information for the kNN classification, involving ant colonies optimization. Experiments have shown that the proposed method effectively reduces the training set, without sacrificing the performance of classification. Our future studies will focus on solving the problem of complexity of the Ant-KNN algorithm to allow its use on larger datasets and its application to intrusion detection to reduce false alarms, especially false positive ones.

References

1. Fabrizio, A.: Fast Condensed Nearest Neighbor Rule. Technical report, Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany (2005)
2. Deneubourg, J.L., Beckers, R., Goss, S.: Trails and U-turns in the Selection of the Shortest Path by the Ant Lasius Niger. Journal of Theoretical Biology 159, 397–415 (1992)
3. Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. IEEE Transaction on Information Theory, IT 13, 21–27 (1967)
4. Marco, D., Luca, M.G.: Ant Colonies for the Traveling Salesman Problem. BioSystems (1997)
5. Marco, D., Thomas, S.: Ant Colony Optimization (2004)

6. Hatem, A.F., Amir, F.A.: A Novel Template Reduction Approach for the K-nearest Neighbor Method. IEEE Transactions on Neural Networks 20(5), 890–896 (2009)
7. Wilfong, G.: Nearest Neighbor Problems. International Journal of Computational Geometry & Applications 2, 383–416 (1992)
8. Gates, G.: The Reduced Nearest Neighbor Rule. IEEE Transactions on Information Theory 18, 431–433 (1972)
9. Aron, S., Deneubourg, J.L., Goss, S., Pasteels, J.M.: Self-organized Shortcuts in the Argentine Ant. Naturwissenschaften 76, 579–581 (1989)
10. Michael, H., Kurt, H.: Tsp - infrastructure for the Traveling Salesperson Problem. Journal of Statistical Software 23(2), 1–21 (2007)
11. Karla, H., Manfred, P.: Salesman Problem. In: Encyclopedia of Operations Research, 2nd edn. (2000)
12. Wu, Q.H., Nikolaidis, K., Goulermas, J.Y.: A Class Boundary Preserving Algorithm for Data Condensation. Pattern Recognition 44, 704–715 (2011)
13. Amal, M., Ahmed, R.B.: Survey of Condensing Nearest Neighbor Techniques. International Journal of Advanced Computer Science and Applications (IJACSA) 2(11), 59–64 (2011)
14. Hart, P.: The Condensed Nearest Neighbor Rule. IEEE Transactions on Information Theory 14, 515–516 (1968)
15. Randall, W., Tony, R.M.: Reduction Techniques for Prototype-based Learning Algorithms. Machine Learning 38(3), 257–286 (2000)
16. Krasimir, G.I., Wu, Y.Q., Venu, G.: Improved K-nearest Neighbor Classification. Pattern Recognition 35, 2311–2318 (2002)

Energy-Efficient Virtual Machine Placement in Data Centers by Genetic Algorithm

Grant Wu[1], Maolin Tang[1], Yu-Chu Tian[1], and Wei Li[2]

[1] School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, QLD 4001, Australia
{m.tang,y64.wu,y.tian}@qut.edu.au
[2] School of Information and Communication Technology,
Central Queensland University, Rockhampton, QLD 4702, Australia
w.li@qut.edu.au

Abstract. Server consolidation using virtualization technology has become an important technology to improve the energy efficiency of data centers. Virtual machine placement is the key in the server consolidation. In the past few years, many approaches to the virtual machine placement have been proposed. However, existing virtual machine placement approaches to the virtual machine placement problem consider the energy consumption by physical machines in a data center only, but do not consider the energy consumption in communication network in the data center. However, the energy consumption in the communication network in a data center is not trivial, and therefore should be considered in the virtual machine placement in order to make the data center more energy-efficient. In this paper, we propose a genetic algorithm for a new virtual machine placement problem that considers the energy consumption in both the servers and the communication network in the data center. Experimental results show that the genetic algorithm performs well when tackling test problems of different kinds, and scales up well when the problem size increases.

Keywords: Virtual machine placement, Server consolidation, Data center, Cloud computing, Genetic algorithm.

1 Introduction

The ever increasing cloud computing has been resulting in ever increasing energy consumption and therefore overwhelming electricity bills for data centers. According to Amazon's estimations, the energy-related costs at its data centers account for 42% of the total operating cost. In addition, the ever increasing energy consumption may lead to dramatically increase in carbon dioxide emissions. So, it is desirable to make every possible effect to reduce the energy consumption in cloud computing.

Server consolidation using visualization technology has become an important technology to improve the energy efficiency of data centers [1]. Virtual machine (VM) placement is the key in the server consolidation. In the past few years,

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 315–323, 2012.

many approaches to various VM placement problems have been proposed. However, existing VM placement approaches do not consider the energy consumption in communication network in the data center. However, the energy consumption in the communication network in a data center is not trivial, and therefore should be considered in VM placement in order to make the data center more energy-efficient.

In this paper, we propose a genetic algorithm (GA) [2] for a new VM placement problem that considers the energy consumption in both the physical servers (PMs) and the communication network in the data center. Experimental results show that the genetic algorithm performs well with various test problems, and scales well when the problem size increases.

The remaining paper is organized as follows: Section 2 formulates the new VM placement problem; Section 3 presents the GA; Section 4 evaluates the performance and scalability of the GA; and finally Section 5 concludes this work.

2 Problem Formulation

Let's define

V	a set of virtual machines
P	a set of physical machines
v_i	a virtual machine in V
v_i^{cpu}	the CPU requirement of v_i
v_i^{mem}	the memory requirement of v_i
p_j	a physical machine in P
p_j^{cpu}	the CPU capacity of p_j
p_j^{mem}	the memory capacity of p_j
$p_j^{w_{cpu}}$	the total CPU workload on p_j
$p_j^{w_{mem}}$	the total memory workload on p_j
V_{p_j}	the set of virtual machines assigned to physical machine p_j

The utilization rate of the CPU in physical server p_j is

$$\mu_j = p_j^{w_{cpu}} / p_j^{cpu} \tag{1}$$

Thus, according to the server energy consumption model defined in [3], the energy consumption of physical server p_j when its CPU usage is μ_j is

$$E(p_j) = k_j \cdot e_j^{max} + (1 - k_j) \cdot e_j^{max} \cdot \mu_j \tag{2}$$

where k_j is the fraction of energy consumed when p_j is idle; e_j^{max} is the energy consumption of physical server p_j when it is fully utilized; and μ_j is the CPU utilization of p_j.

It is assumed that the communication network topology of the data center is a typical three-tier one as shown in Fig. 1 [4]. The VMs in the data center may communicate with each other through the communication devices, such as switches, which also consume a non-trivial amount of energy and it has been shown that this energy consumption is largely independent of the load through

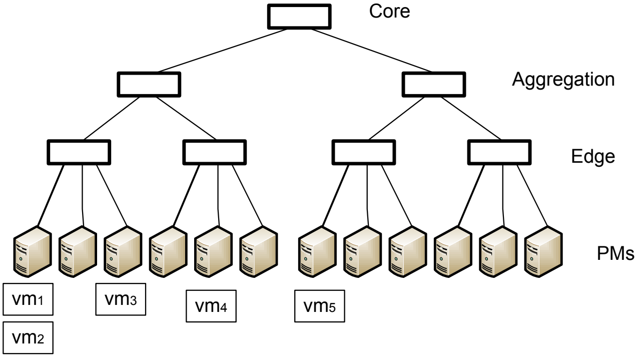


Fig. 1. The communication network of a data center

the communication devices [5]. Thus, we use the following method to approximate the energy consumption in the communication network in the data center.

We categorize the communication between a pair of VMs into four types: The first type is that the pair of VMs are on the same PM. The communication between vm_1 and vm_2 in Fig. 1 is an instance of the first type. The second type is that the pair of VMs are placed on two different PMs, but under the same edge. The communication between vm_1 and vm_3 in Fig. 1 is an example of the second type. The third type is that the pair of VMs are placed on two different PMs under different edges, but under the same aggregation. The communication between vm_3 and vm_4 in Fig. 1 is an example of the third type. The fourth is that the pair of VMs are placed on two different PMs under different edges and different aggregations. The communication between vm_4 and vm_5 in Fig. 1 is an example of the fourth type.

The first type of communication does not use any network communication device; the second type of communication uses one network communication device; the third communication involves in three network communication devices; and the fourth type of communication is done through five network communication devices. Therefore, the energy consumptions incurred by the four types of communication are different. In fact, the first type of communication does not incur any energy consumption in the communication network; the energy consumption of the second type communication is less than that of the third type, which is in turn less than that of the fourth type as the more network communication devices are used, the more energy is consumed in the communication network.

Let C_1, C_2, C_3 and C_4 be the sets of VM pairs between which there exists communication and the type communication belong to the first, second, third and fourth, respectively; and

$$C = C_1 \cup C_2 \cup C_3 \cup C_4 \tag{3}$$

For each communication $c \in C$, the energy consumption for transferring a unit of data is

$$e(c) = \begin{cases} 0, & \text{if } c \in C_1; \\ e_2, & \text{if } c \in C_2; \\ e_3, & \text{if } c \in C_3; \\ e_4, & \text{if } c \in C_4; \end{cases} \tag{4}$$

Let $l(c)$ be the amount of data that need to be transferred on the communication c. Then, the network energy consumption for transferring $l(c)$ units of data is

$$E(c) = e(c) * l(c) \tag{5}$$

the virtual machine placement problem is to assign each virtual machine in V onto a physical machine in P, such that

$$\sum_{p_j \in P} E(p_j) + \sum_{c \in C} E(c) \tag{6}$$

is minimized subject to

$$\bigcup_{p_j \in P} V_{p_j} = V \tag{7}$$

$$V_{p_i} \bigcap_{p_i \neq p_j} V_{p_j} = \emptyset \tag{8}$$

$$p_j^{w_{cpu}} = \sum_{v_i \in V_{p_j}} v_i^{cpu} \leq p_j^{cpu} \tag{9}$$

$$p_j^{w_{mem}} = \sum_{v_i \in V_{p_j}} v_i^{mem} \leq p_j^{mem} \tag{10}$$

Constraints (7) and (8) make sure that each virtual machine will be assigned to one and only one physical machine; constraints (9) and (10) guarantee that the total CPU workload and the total memory on physical machine p_j will not exceed the CPU capacity and the memory capacity, respectively.

3 Genetic Algorithm

This section entails the GA for the VM placement problem. It discusses in detail the encoding scheme, genetic operators and fitness function of the GA as well as the description of the GA.

3.1 Encoding Scheme

A chromosome in this GA consists of $|V|$ genes, each of which stands for a virtual machine. The value of a gene is a positive integer between 1 and $|P|$, representing the physical machine where the virtual machine is allocated. Fig. 2 shows a example VM placement and its corresponding chromosome.

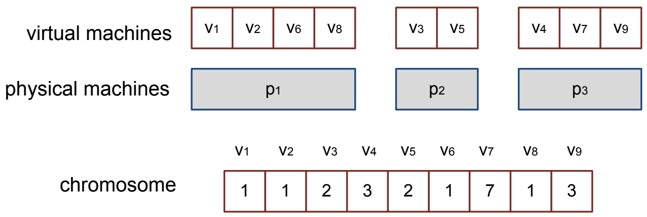


Fig. 2. An example of VM placement and its corresponding chromosome

3.2 Crossover

Since the length of chromosome is potentially long, linkage is a potential problem that should be considered. Because of this consideration, the GA adopts a biased uniform crossover operator, which is described in Algorithm 1.

Algorithm 1: Biased Uniform Crossover

```
Input  : two parent chromosomes, $C^i = x_1^i x_2^i \cdots x_n^i$ and $C^j = x_1^j x_2^j \cdots x_n^j$
Output: one child chromosome, $C^k = x_1^k x_2^k \cdots x_n^k$
$f^i \leftarrow fitness(C^i)$;
$f^j \leftarrow fitness(C^j)$;
for $q = 1$ to $n$ do
    randomly generate a real value between 0 and 1, $r$;
    if $r < f^i/(f^i + f^j)$ then
        $x_q^k \leftarrow x_q^i$;
    end
    else
        $x_q^k \leftarrow x_q^j$;
    end
end
output $C^k$.
```

3.3 Mutation

The mutation operator simply randomly picks up a gene in the chromosome and inverts the value of the chosen gene. Algorithm 2 shows how the mutation operator works.

3.4 Fitness Function

The fitness of an individual x in the population of the GA is defined in Eq. 11 below:

$$fitness(x) = \begin{cases} E^{min}/E(x), & \text{if } x \text{ is feasible;} \\ E^{min}/(E(x) + E^{max}), & \text{otherwise.} \end{cases} \tag{11}$$

Algorithm 2: Mutation

```
Input  : a chromosome, $C = x_1 x_2 \cdots x_n$
Output: a mutated chromosome, $C' = x'_1 x'_2 \cdots x'_n$
$C' \leftarrow C$;
randomly generate a virtual machine $i$, where $1 \leq i \leq |V|$;
randomly generate a physical machine $p$, where $1 \leq p \leq |P|$;
replace $x'_i \leftarrow p$;
output $C'$.
```

where E^{min} is a lower boundary of the total energy consumption, E^{max} is an upper boundary of the total energy consumption, and $E(x)$ is the total energy consumption when VM placement x is adopted.

The fitness function penalizes a solution that violates any of those constraints, and make sure that the fitness value of any infeasible solution is less than that of any feasible solution and that the less energy consumption and the greater the fitness value is.

3.5 The Description of the GA

Algorithm 3 is a high-level description of the GA.

Algorithm 3: The GA

```
generate a population of $PopSize$ individuals, $Pop$;
find the best individual in $Pop$;
while the termination condition is not true do
    for each individual $x$ in $Pop$ do
        calculate its fitness value $f(x)$;
    end
    for each individual in $Pop$ do
        use the roulette selection to select another individual to pair up;
    end
    for each pair of parents do
        probabilistically use the biased uniform crossover operator to produce
        an offspring;
    end
    for each individual in $P$ do
        probabilistically apply the mutation operator the individual;
    end
    find the best individual in $Pop$;
    if the best individual in $Pop$ is better than the current best individual then
        replace the current best individual with the new best individual;
    end
end
decode the best individual and output it.
```

4 Evaluation

The GA has been implemented in Java. Since there are no benchmarks available for the new VM placement problem, we have to randomly generate test problems to test the GA. We use a set of experiments to evaluate the proposed GA with respect to performance and scalability. Table 1 shows the characteristics of those randomly generated test problems:

Table 1. Characteristics of test problems

Test problem	VM (#)	PM(#)
1	100	20
2	200	40
3	300	60
4	400	80
5	500	100

In all the experiments, the population size of the GA was 200, the probabilities for crossover and mutation were 0.5 and 0.1, respectively, and the termination condition was “no improvement in the best solution for 20 generations".

In these randomly generated test problems, the VMs' CPU and memory requirements were randomly generated and the values were both in $[300, 3000]$, and the PMs' CPU and memory capacities were both randomly picked up from $\{1000, 1500, \cdots, 55000\}$. The parameters about the communication network were: $e_2 = 1$; $e_3 = 3$; and $e_4 = 5$. The amount of data need to be transferred between each pair of VMs in C was randomly generated and the value was a whole number between 1 and 9 (units). The parameters about the servers in the data center were: $k_1 = k_2 = \cdots = k_{|P|} = 0.7$.

For each of the randomly generated test problems, we used the GA to solve it. Considering the stochastic nature of the GA, we repeated the experiments 10 times, and recorded the solutions and computation times. Since it was difficult or impossible to know the optimal solutions to those test problems and therefore to know the quality of the solutions generated by the GA, we implemented an First Fit Decreasing (FFD) algorithm in Java, and used it to solve those test problems. The FFD algorithm is one the most popular heuristic algorithms for bin packing problems. Since VM placement problems can be easily transformed into a bin packing problem, the FFD algorithm is often used to tackle VM placement problems [6]. Since the FFD algorithm is a deterministic one, we only ran it once for each of the test problems. We evaluated the performance of the GA by comparing the quality of the solutions generated by the GA with the quality of the solutions produced by the FFD-based heuristic algorithm. Table 2 shows the experimental results.

It can be seen from the experimental results in Table 2 that the solutions produced by the GA are significantly better than those produced by the FFD. On average the solutions produced by the GA are 3.5%-23.5% better than those produced by the FFD.

Table 2. Comparison of the performance of the GA and the performance of the FFD

Test	FFD	GA			
Problem	Energy (watts)	Energy (watts)	SD	Time (seconds)	SD
100	12746.46	10317.73	763.16	51.63	19.40
200	24862.72	22525.48	1322.34	357.58	75.28
300	42035.96	37555.60	1849.23	1011.44	286.42
400	56223.20	51796.05	1620.96	2139.52	507.66
500	70320.00	67912.29	1645.19	3256.46	518.43

In terms of computation time, the FFD took less than 1 millisecond to solve any of the five test problems. The computation time of the GA increased with the number f VMs and the number of PMs. It was observed that the computation time of the GA increased linearly with the product of the number of VMs and the number of PMs. Fig. 3 visualizes the observation. Given that this virtual machine placement problem is a static optimization problem, the computation time and the scalability of the GA are acceptable.

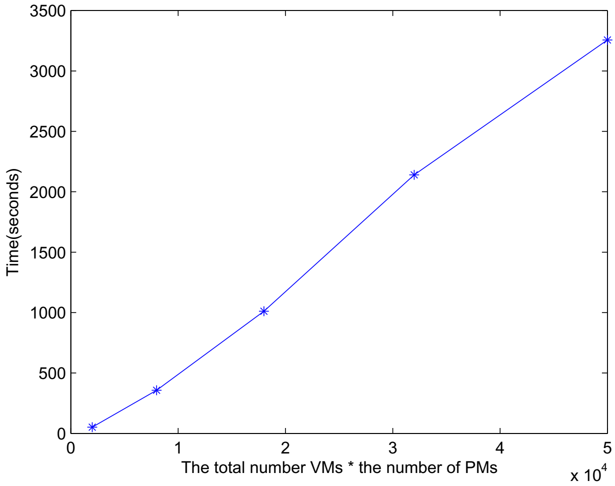


Fig. 3. The scalability of our GA

5 Conclusion

In this paper we have identified and formulated a new VM placement problem. The new VM placement problem considers not only the energy consumption in those physical servers in a data center, but also the energy consumption in the communication network of the data center. In addition, this paper has proposed a GA for the new VM placement problem. The GA has been implemented and evaluated by experiments. Experimental results have shown that the GA always generates a significantly better solution than the FFD-based algorithm for the VM placement problem.

In this work we used simple energy consumption models to calculate the energy consumptions in the physical servers and the communication network of a data center. However, our GA is independent from those energy consumption models. Thus, in the future we will use more accurate energy consumption models when they are available.

References

1. Meng, X., Pappas, V., Zhang, L.: Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement. In: Proceeding of IEEE International Conference on Computer Communications, pp. 1–9 (2010)
2. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning. Addison Wesley (1989)
3. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing. Future Generation Computer Systems 28(5), 755–768 (2012)
4. Benson, T., Akella, A., Maltz, D.A.: Network Traffic Characteristics of Data Centers in the Wild. In: Proceedings of the 10th Annual Conference on Internet Measurement, pp. 267–280 (2010)
5. Mahadevan, P., Sharma, P., Banerjee, S., Ranganathan, P.: Energy Aware Network Operations. In: Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), pp. 25–30 (2009)
6. Xu, J., Fortes, J.A.B.: Multi-objective Virtual Machine Placement in Virtualized Data Center Environments. In: Proceeding of IEEE/ACM International Conference on Green Computing and Communications, pp. 179–188 (2010)

A Contextual-Bandit Algorithm for Mobile Context-Aware Recommender System

Djallel Bouneffouf, Amel Bouzeghoub, and Alda Lopes Gançarski

Department of Computer Science, Télécom SudParis, UMR CNRS Samovar,
91011 Evry Cedex, France
{Djallel.Bouneffouf,Amel.Bouzeghoub,
Alda.Gancarski}@it-sudparis.eu

Abstract. Most existing approaches in Mobile Context-Aware Recommender Systems focus on recommending relevant items to users taking into account contextual information, such as time, location, or social aspects. However, none of them has considered the problem of user's content evolution. We introduce in this paper an algorithm that tackles this dynamicity. It is based on dynamic exploration/exploitation and can adaptively balance the two aspects by deciding which user's situation is most relevant for exploration or exploitation. Within a deliberately designed offline simulation framework we conduct evaluations with real online event log data. The experimental results demonstrate that our algorithm outperforms surveyed algorithms.

Keywords: Recommender system, Machine learning, Exploration/exploitation dilemma, Artificial intelligence.

1 Introduction

Mobile technologies have made access to a huge collection of information, anywhere and anytime. In particular, most professional mobile users acquire and maintain a large amount of content in their repository. Moreover, the content of such repository changes dynamically, undergoes frequent insertions and deletions. In this sense, recommender systems must promptly identify the importance of new documents, while adapting to the fading value of old documents. In such a setting, it is crucial to identify interesting content for users. This problem has been addressed in recent research in the Mobile Context-Aware Recommender Systems (MCRS) area [2, 4, 5, 14]. Most of these approaches are based on the user computational behavior and his surrounding environment. Nevertheless, they do not tackle the dynamicity of the user's content problem. The bandit algorithm is a well-known solution that addresses this problem as a need for balancing exploration/exploitation (exr/exp) tradeoff. A bandit algorithm B exploits its past experience to select documents that appear more frequently. Besides, these seemingly optimal documents may in fact be suboptimal, because of the imprecision in B's knowledge. In order to avoid this undesired case, B has to explore documents by choosing seemingly suboptimal documents so as to gather more information about them. Exploitation can decrease short-term user's satisfaction since some suboptimal documents may be chosen. However, obtaining

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 324–331, 2012.

information about the documents' average rewards (i.e., exploration) can refine B's estimate of the documents' rewards and in turn increases long-term user's satisfaction. Clearly, neither a purely exploring nor a purely exploiting algorithm works well, and a good tradeoff is needed. One classical solution to the multi-armed bandit problem is the ε-greedy strategy [12]. With the probability 1-ε, this algorithm chooses the best documents based on current knowledge; and with the probability ε, it uniformly chooses any other documents uniformly. The ε parameter controls essentially the exp/exr tradeoff between exploitation and exploration. One drawback of this algorithm is that it is difficult to decide in advance the optimal value. Instead, we introduce an algorithm named Contextual-ε-greedy that achieves this goal by balancing adaptively the exp/exr tradeoff according to the user's situation. This algorithm extends the ε-greedy strategy with an update of the exr/exp-tradeoff by selecting suitable user's situations for either exploration or exploitation.

The rest of the paper is organized as follows. Section 2 gives the key notions used throughout this paper. Section 3 reviews some related works. Section 4 presents our MCRS model and describes the algorithms involved in the proposed approach. The experimental evaluation is illustrated in Section 5. The last section concludes the paper and points out possible directions for future work.

2 Key Notions

In this section, we briefly sketch the key notions that will be of use in this paper.

The User's Model: The user's model is structured as a case based, which is composed of a set of situations with their corresponding user's preferences, denoted U = {(S^i; UP^i)}, where S^i is the user's situation and UP^i its corresponding user's preferences.

The User's Preferences: The user's preferences are deduced during the user's navigation activities, for example the number of clicks on the visited documents or the time spent on a document. Let UP be the preferences submitted by a specific user in the system at a given situation. Each document in UP is represented as a single vector d=(c_1,...,c_n), where c_i (i=1, .., n) is the value of a component characterizing the preferences of d. We consider the following components: the total number of clicks on d, the total time spent reading d and the number of times d was recommended.

Context: A user's context *C* is a multi-ontology representation where each ontology corresponds to a context dimension $C=(O_{Location}, O_{Time}, O_{Social})$. Each dimension models and manages a context information type. We focus on these three dimensions since they cover all needed information. These ontologies are described in [1].

Situation: A situation is an instantiation of the user's context. We consider a situation as a triple $S = (O_{Location}.x_i, O_{Time}.x_j, O_{Social}.x_k)$ where x_i, x_j and x_k are ontology concepts or instances. Suppose the following data are sensed from the user's mobile phone: the GPS shows the latitude and longitude of a point "48.89, 2.23"; the local time is "Oct_3_12:10_2012" and the calendar states "meeting with Paul Gerard". The corresponding situation is: *S=("48.89,2.23","Oct_3_12:10_2012","Paul_Gerard").* To build a more abstracted situation, we interpret the user's behavior from this low-level multimodal sensor data using ontologies reasoning means. For example, from *S*, we obtain the following situation: *Meeting=(Restaurant, Work_day, Financial_client).* Among the set of captured situations, some of them are characterized as High-Level Critical Situations.

High-Level Critical Situations (HLCS): A HLCS is a class of situations where the user needs the best information that can be recommended by the system, for instance, during a professional meeting. In such a situation, the system must exclusively perform exploitation rather than exploration-oriented learning. In the other case, where the user is for instance using his/her information system at home, on vacation with friends, the system can make some exploration by recommending some information ignoring his/her interest. The HLCS are predefined by the domain expert. In our case we conduct the study with professional mobile users, which is described in detail in Section 5. As examples of HLCS, we can find S1 = (*restaurant, midday, client*) or S2= (*company, morning, manager*).

3 Related Work

We refer, in the following, recent recommendation techniques that tackle the problem of making dynamic exr/exp (bandit algorithms). Existing works considering the user's situation in recommendation are not considered in this section, refer to [1] for further information.

Very frequently used in reinforcement learning to study the exr/exp tradeoff, the multi-armed bandit problem was originally described by Robbins [11]. The *ε-greedy* is one of the most used strategy to solve the bandit problem and was first described in [10]. The *ε-greedy* strategy chooses a random document with epsilon-frequency (ε), and chooses the document with the highest estimated mean otherwise. The estimation is based on the rewards observed thus far. ε must be in the interval [0, 1] and its choice is left to the user. The first variant of the *ε-greedy* strategy is what [6, 10] refer to as the *ε-beginning* strategy. This strategy makes exploration all at once at the beginning. For a given number I of iterations, documents are randomly pulled during the εI first iterations; during the remaining $(1-\varepsilon)I$ iterations, the document of highest estimated mean is pulled. Another variant of the *ε-greedy* strategy is what [10] calls the *ε-decreasing*. In this strategy, the document with the highest estimated mean is always pulled except when a random document is pulled instead with ε_i frequency, where $\varepsilon_i = \{\varepsilon_0/ i\}$, $\varepsilon_0 \in]0,1]$ and i is the index of the current round. Besides *ε-decreasing*, four other strategies presented [3]. Those strategies are not described here because the experiments done by [3] seem to show that *ε-decreasing* is always as good as the other strategies. Compared to the standard multi-armed bandit problem with a fixed set of possible actions, in MCRS, old documents may expire and new documents may frequently emerge. Therefore it may not be desirable to perform the exploration all at once at the beginning as in [6] or to decrease monotonically the effort on exploration as the decreasing strategy in [10].

As far as we know, no existing works address the problem of exr/exp tradeoff in MCRS. However few research works are dedicated to study the contextual bandit problem on recommender systems, where they consider the user's behavior as the context of the bandit problem. In [13], the authors extend the ε-greedy strategy by dynamically updating the ε exploration value. At each iteration, they run a sampling procedure to select a new ε from a finite set of candidates. The probabilities associated to the candidates are uniformly initialized and updated with the Exponentiated Gradient (EG) [7]. This updating rule increases the probability of a candidate *ε* if it leads to a user's click. Compared to both *ε-beginning* and *ε-decreasing*, this technique gives better results. In [9], authors model the recommendation as a contextual bandit problem. They propose an approach in which a learning algorithm sequentially selects documents to serve users based on their behavior. To maximize the total number of user's clicks, this work proposes LINUCB algorithm that is computationally efficient.

As shown above, none of the mentioned works tackles both problems of exr/exp dynamicity and user's situation consideration in the exr/exp strategy. This is precisely what we intend to do with our approach. Our intuition is that, considering the criticality of the situation when managing the exr/exp-tradeoff, improves the result of the MCRS. This strategy achieves high exploration when the current user's situation is not critical and achieves high exploitation in the inverse case.

4 MCRS Model

In our recommender system, the recommendation of documents is modeled as a contextual bandit problem including user's situation information [8]. Formally, a bandit algorithm proceeds in discrete trials *t* = *1…T*. For each trial *t*, the algorithm performs the following tasks:

Task 1: Let S^t be the current user's situation, and *PS* the set of past situations. The system compares S^t with the situations in *PS* in order to choose the most similar one, S^p:

$$S^p = \underset{S^c \in PS}{\arg\max} \left(sim(S^t, S^c) \right) \tag{1}$$

The semantic similarity metric is computed by:

$$sim(S^t, S^c) = \sum_j \alpha_j \cdot sim_j \left(x_j^t, x_j^c \right) \tag{2}$$

where sim_j is the similarity metric related to dimension *j* between two concepts x_j^t and x_j^c; α_j is the weight associated to dimension *j* (during the experimental phase, α_j has a value of 1 for all dimensions). This similarity depends on how closely x_j^c and x_j^c are related in the corresponding ontology. We use the same similarity measure as [15] defined by:

$$sim_j \left(x_j^t, x_j^c \right) = 2 * \frac{deph(LCS)}{(deph(x_j^c) + deph(x_j^t))} \tag{3}$$

where LCS is the *Least Common Subsumer* of x_j^t and x_j^c, and *deph* is the number of nodes in the path from the node to the ontology root.

Task 2: Let D be the document collection and $D_p \in D$ the set of documents recommended in situation S^p. After retrieving S^p, the system observes the user's behavior when reading each document $d_p \in D_p$. Based on observed rewards, the algorithm chooses document d_p with the greater reward r_p.

Task 3: After receiving the user's reward, the algorithm improves its document-selection strategy with the new observation: in situation S^t, document d_p obtains a reward r_t.

When a document is presented to the user and this one selects it by a click, a reward of 1 is incurred; otherwise, the reward is 0. The reward of a document is precisely its Click Through Rate (CTR). The CTR is the average number of clicks on a document by recommendation.

4.1 The ε-Greedy Algorithm

The *ε-greedy* algorithm recommends a predefined number of documents N selected using the following equation:

$$d_i = \begin{cases} \operatorname{argmax}_{UC}(getCTR(d)) & if\,(q > \varepsilon) \\ Random(UC) & otherwise \end{cases} \tag{4}$$

where $i \in \{1,\ldots N\}$, $UC=\{d_1,\ldots,d_P\}$ is the set of documents corresponding to the user's preferences; *getCTR()* computes the CTR of a given document; *Random()* returns a random element from a given set, allowing to perform exploration; q is a random value uniformly distributed over [0, 1] which defines the exr/exp tradeoff; ε is the probability of recommending a random exploratory document.

4.2 The Contextual-ε-Greedy Algorithm

To improve the adaptation of the *ε-greedy* algorithm to HLCS situations, the *contextual-ε-greedy* algorithm compares the current user's situation S^t with the HLCS class of situations. Depending on the similarity between the S^t and its most similar situation $S^m \in$ HLCS, being B the similarity threshold (this metric is discussed below), two scenarios are possible:

(1) If $sim(S^t, S^m) \geq B$, the current situation is critical; the *ε-greedy* algorithm is used with $\varepsilon=0$ (exploitation) and S^t is inserted in the HLCS class of situations.
(2) If $sim(S^t, S^m) < B$, the current situation is not critical; the *ε-greedy* algorithm is used with $\varepsilon>0$ (exploration) computed as indicated in Eq.5.

$$\varepsilon = \begin{cases} 1-\left(\dfrac{sim(S^t, S^m)}{B}\right) & if\,(sim(S^t, S^m)) < B \\ 0 & otherwise \end{cases} \tag{5}$$

To summarize, the system does not make exploration when the current user's situation is critical; otherwise, the system performs exploration. In this case, the degree of exploration decreases when the similarity between S^t and S^m increases.

5 Experimental Evaluation

In order to empirically evaluate the performance of our approach, and in the absence of a standard evaluation framework, we propose an evaluation framework based on a diary set of study entries. The main objectives of the experimental evaluation are: (1) to find the optimal threshold B value described in Section 4.2 and (2) to evaluate the performance of the proposed algorithm (*contextual-ε-greedy*). In the following, we describe our experimental datasets and then present and discuss the obtained results.

We have conducted a diary study with the collaboration of the French software company Nomalys[1]. This company provides a history application, which records the time, current location, social and navigation information of its users during their application use. The diary study has taken 18 months and has generated 178 369 diary situation entries. Each diary situation entry represents the capture, of contextual time, location and social information. For each entry, the captured data are replaced with more abstracted information using time, spatial and social ontologies [1]. From the diary study, we have obtained a total of 2 759 283 entries concerning the user's navigation, expressed with an average of 15.47 entries per situation.

In order to set out the threshold similarity value, we use a manual classification as a baseline and compare it with the results obtained by our technique. So, we take a random sampling of 10% of the situation entries, and we manually group similar situations; then we compare the constructed groups with the results obtained by our similarity algorithm, with different threshold values.

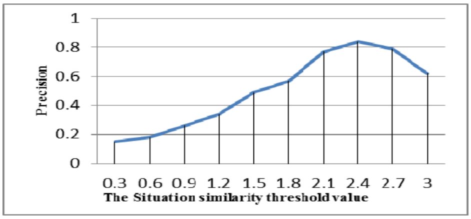


Fig. 1. Effect of B threshold value on the similarity precision

Fig. 1 shows the effect of varying the threshold situation similarity parameter B in the interval [0, 3] on the overall precision. Results show that the best performance is obtained when B has the value 2.4 achieving a precision of 0.849. Consequently, we use the optimal threshold value B = 2.4 for testing our MCRS.

[1] Nomalys is a company that provides a graphical application on Smartphones allowing users to access their company's data.

To test the proposed *contextual-ε-greedy* algorithm, we firstly have collected 3000 situations with an occurrence greater than 100 to be statistically meaningful. Then, we have sampled 10000 documents that have been shown on any of these situations. The testing step consists of evaluating the algorithms for each testing situation using the average CTR. The average CTR for a particular iteration is the ratio between the total number of clicks and the total number of displays. Then, we calculate the average CTR over every 1000 iterations. The number of documents (N) returned by the recommender system for each situation is 10 and we have run the simulation until the number of iterations reaches 10000, which is the number of iterations where all algorithms have converged. In the first experiment, in addition to a pure exploitation baseline, we have compared our algorithm to the algorithms described in the related work (Section 3): *ε-greedy*; *ε-beginning*, *ε-decreasing* and EG. In Fig. 2, the horizontal axis is the number of iterations and the vertical axis is the performance metric.

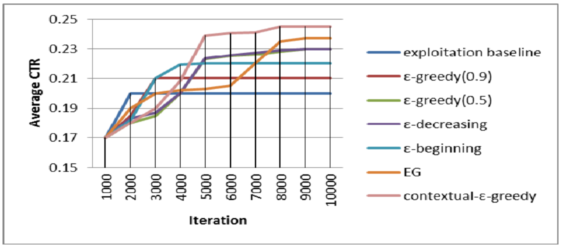


Fig. 2. Average CTR for exr/exp algorithms

We have parameterized the different algorithms as follows: *ε-greedy* was tested with two parameter values: 0.5 and 0.9; *ε-decreasing* and *EG* use the same set *{ε$_i$ = 1-0.01 * i, i = 1,...,100}*; *ε-decreasing* starts using the highest value and reduces it by 0.01 every 100 iterations, until it reaches the smallest value. Overall tested algorithms have better performance than the baseline. However, for the first 2000 iterations, with pure exploitation, the exploitation baseline achieves a faster increase convergence. But in the long run, all exr/exp algorithms improve the average CTR at convergence. We have several observations regarding the different exr/exp algorithms. For the *ε-decreasing* algorithm, the converged average CTR increases as the ε decreases (exploitation augments). For the ε-*greedy(0.9)* and ε-*greedy(0.5),* even after convergence, the algorithms still give respectively 90% and 50% of the opportunities to documents having low average CTR, which decreases significantly their results. While the *EG* algorithm converges to a higher average CTR, its overall performance is not as good as *ε-decreasing*. Its average CTR is low at the early step because of more exploration, but does not converge faster. The *contextual-ε-greedy* algorithm effectively learns the optimal ε; it has the best convergence rate, increases the average CTR by a factor of 2 over the baseline and outperforms all other exr/exp algorithms. The improvement comes from a dynamic tradeoff between exr/exp, controlled by the critical situation (HLCS) estimation. At the early stage, this algorithm takes full advantage of exploration without wasting opportunities to establish good results.

6 Conclusion

In this paper, we study the problem of exploitation and exploration in mobile context-aware recommender systems and propose a novel approach that balances adaptively exr/exp regarding the user's situation. In order to evaluate the performance of the proposed algorithm, we compare it with other standard exr/exp strategies. The experimental results demonstrate that our algorithm performs better on average CTR in various configurations. In the future, we plan to evaluate the scalability of the algorithm on-board a mobile device and investigate other public benchmarks.

References

1. Bouneffouf, D., Bouzeghoub, A., Gançarski, A.L.: Following the User's Interests in Mobile Context-Aware Recommender Systems. In: AINA Workshops, Fukuoka, Japan, pp. 657–662 (2012)
2. Adomavicius, G., Mobasher, B., Ricci, F., Alexander, T.: Context-Aware Recommender Systems. AI Magazine 32(3), 67–80 (2011)
3. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite Time Analysis of the Multiarmed Bandit Problem. Machine Learning 2, 235–256 (2002)
4. Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: Context Relevance Assessment and Exploitation in Mobile Recommender Systems. Personal and Ubiquitous Computing, 1–20 (2011)
5. Bellotti, V., Begole, B., Chi, E.H., Ducheneaut, N., Fang, J., Isaacs, E.: Activity-Based Serendipitous Recommendations with the Magitti Mobile Leisure Guide. In: Proceedings on the Move, pp. 1157–1166. ACM, New York (2008)
6. Even-Dar, E., Mannor, S., Mansour, Y.: PAC Bounds for Multi-armed Bandit and Markov Decision Processes. In: Kivinen, J., Sloan, R.H. (eds.) COLT 2002. LNCS (LNAI), vol. 2375, p. 255. Springer, Heidelberg (2002)
7. Kivinen, J., Warmuth, M.K.: Exponentiated Gradient versus Gradient Descent for Linear Predictors. Information and Computation 132 (1995)
8. Langford, J., Zhang, T.: The Epoch-greedy Algorithm for Contextual Multi-armed Bandits. In: Advances in Neural Information Processing Systems (2008)
9. Li, L., Wei, C., Langford, J.: A Contextual-Bandit Approach to Personalized News Document Recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 661–670. ACM, New York (2010)
10. Mannor, S., Tsitsiklis, J.N.: The Sample Complexity of Exploration in the Multi-Armed Bandit Problem. In: Computational Learning Theory, pp. 255–270 (2003)
11. Robbins, H.: Some Aspects of the Sequential Design of experiments. Bulletin of the American Mathematical Society 58, 527–535 (1952)
12. Watkins, C.: Learning from Delayed Rewards. Ph.D. thesis. Cambridge University (1989)
13. Li, W., Wang, X., Zhang, R., Cui, Y., Mao, J., Jin, R.: Exploitation and Exploration in a Performance Based Contextual Advertising System. In: Proceedings of the International Conference on Knowledge discovery and data mining, pp. 27–36. ACM, New York (2010)
14. Sohn, T., Li, K.A., Griswold, W.G., Hollan, J.D.: A Diary Study of Mobile Information Needs. In: Proceedings of the 26th Annual SIGCHI Conference on Human Factors in Computing, pp. 433–442. ACM, Florence (2008)
15. Wu, Z., Palmer, M.: Verb Semantics and Lexical Selection. In: Proceedings of the 32nd Meeting of the Association for Computational Linguistics, pp. 133–138 (1994)

Multi-source Transfer Learning with Multi-view Adaboost

Zhijie Xu and Shiliang Sun

Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, P.R. China
zjxu09@gmail.com, slsun@cs.ecnu.edu.cn

Abstract. Transfer learning, which is one of the most important research directions in machine learning, has been studied in various fields in recent years. In this paper, we combine the theories of multi-source and multi-view learning into transfer learning and propose a new algorithm named Multi-source Transfer Learning with Multi-view Adaboost (MsTL-MvAdaboost). Different from many previous works on transfer learning, in this algorithm, we not only use the labeled data from several source tasks to help learn one target task, but also consider how to transfer them in different views synchronously. We regard all the source and target tasks as a collection of several constituent views and each of these tasks can be learned from different views. Experimental results also validate the effectiveness of our proposed approach.

Keywords: Transfer learning, Multi-source learning, Multi-view learning, Adaboost, Supervised learning.

1 Introduction

Traditional machine learning depends on the availability of a large number of data from a single task to train an effective model. However, researchers often confront the situations that there are not enough data available and they have to resort to data from other tasks (source tasks) to aid the learning of the target task. Due to the above reasons, transfer learning [1,2] begins to catch more attention in recent years. In this paper, in order to promote the effectiveness of transfer learning, we incorporate the adaboost algorithm [3] into it, together with multi-view learning [4]. In addition, because of the different distributions between the target task and source tasks, not all of the knowledge from source tasks can be reused in the target task and some of them may lead to negative transfer [5]. For the purpose of avoiding this problem, we can make use of multi-source learning [6] simultaneously.

Sometimes, although some data in source tasks are unsuitable for the target task, there still may exist some other data that can be useful and helpful for the target task. To find out this kind of data, we employ the adaboost algorithm by voting on every datum. In addition, different feature sets of data can exhibit a common underlying structure. Therefore, through learning the same task from

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 332–339, 2012.

diverse views, we can get various kinds of knowledge which can take different effects on the model.

In this paper, on the basis of the algorithm Multi-view Transfer Learning with Adaboost (MV-TLAdaboost, MV-TLAda for short) [7], we present a new algorithm, Multi-source Transfer Learning with Multi-view Adaboost (MsTL-MvAdaboost, MsTL-MvAda for short) by combining the advantages of multi-source learning into multi-view transfer learning algorithm. The function of it can be understood from the following two points. Firstly, depending on the algorithm MV-TLAdaboost, we can judge whether one datum from the source task can be reused in the target task effectively. Secondly, with the help of multi-source learning, we can prevent the negative transfer and promote the effectiveness of transfer learning.

2 Adaboost

Adaboost is a supervised learning technique for incrementally building linear combinations of weak learners to generate a strong predicative model. In this algorithm, we regard the input data set as $X = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ where x_i belongs to a domain D and y_i belongs to the class label set $Y = \{0, 1\}$. Then, we supply a weight set $W = \{w_1, w_2, \cdots, w_n\}$ for all the samples, and initialize them by $1/n$, where n is the size of X. Next, we start the iteration for T times with the distribution P of samples. Following this, in every iteration, the algorithm comes to one weaker learner h_t for the training step. Moreover, the weight set W needs to be updated by the parameter β_t which is composed by the error rate ε_t of the weaker learner h_t. Finally, the set of weaker learners $H = \{h_1, h_2, \cdots, h_T\}$ are combined by weighted majority voting into the final learner. Details can be seen in the table named Algorithm Adaboost below.

```
                         Algorithm Adaboost
Input:
I. Sequence of $n$ labeled examples $X = (x_1, y_1), \cdots, (x_n, y_n)$.
II. Distribution $D$ over the $n$ examples.
III. Integer T specifying number of iterations.
Initialize the weight vector: $w_1^i = D(i)\; for\, i = 1, \cdots, n$.
For $t = 1, 2, \cdots, T$
    1. Set $P^t = \frac{W^t}{\sum_{i=1}^n w_i^n}$
    2. Call WeakLearn with distribution $P^t$, get back a hypothesis $h_t : X \rightarrow \{0, 1\}$
    3. Calculate the error of $h_t : \varepsilon_t = \sum_{i=1}^n p_i^t |h_t(x_i) - y_i|$
    4. Set $\beta_t = \frac{\varepsilon_t}{(1-\varepsilon_t)}$
    5. Set the new weights vector to be:   $w_i^{t+1} = w_i^t \beta_t^{1-|h_t(x_i)-y_i|}$
End of For
Output the hypothesis:
    $h_f(x) = \begin{cases} 1 \;\; if \quad \sum_{t=1}^T (\log 1/\beta_t) h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \log 1/\beta_t \\ 0 \;\; \text{otherwise} \end{cases}$
```

3 The Proposed Method

Preliminary knowledge on adaboost and transfer learning can be found in [4,7]. Here, we give our proposed algorithm, MsTL-MvAdaboost.

3.1 Overview

As far as we know, there are no previous works focusing on the study of combining multi-source and multi-view learning into transfer learning as a whole. As a result, we design a new algorithm, MsTL-MvAdaboost, to integrate them. Details can be seen in the table named Algorithm MsTL-MvAdaboost below.

3.2 MsTL-MvAdaboost

MV-TLAdaboost is one kind of transfer learning algorithm with only one source task, which means it is intrinsically vulnerable to negative transfer. However, in our proposed algorithm, MsTL-MvAdaboost, for the purpose of avoiding the problem caused by negative transfer and improving the performance of transfer learning, we assume that multiple source tasks can be obtained simultaneously.

Generally speaking, not all the source tasks will be similar enough to the target task. As a result, in order to judge which source task is the most suitable one to help learn the target task, the inner loop computes m pairs of weaker learners $\{h_t^{k_{V1}}, h_t^{k_{V2}}\}$ from m different combined data sets $X \cup S_k$ in every iteration and calculates their error rates ε_t^k about the target data set X at the same time. These actions can be seen from step 2 to step 4. Then, in step 5, with the help of these error rates, we regard the source data set S_k and its weaker learners $\{h_t^{k_{V1}}, h_t^{k_{V2}}\}$ which come to the minimum error rate ε_t^k as the final choice in this iteration. Details can be seen in (1).

$$\begin{cases} \{h_t^{V1}, h_t^{V2}\} = \{h_t^{k_{V1}}, h_t^{k_{V2}}\} \\ \varepsilon_t = \varepsilon_t^k \\ S_t = S_k \end{cases} \tag{1}$$

Moreover, the same as MV-TLAdaboost, it needs to notice that every source data set in our algorithm will generate two weaker learners, $h_t^{k_{V1}}$ and $h_t^{k_{V2}}$ from two views V1 and V2, which are formed at the beginning of the whole algorithm. In addition, in step 4, our algorithm assumes that one sample will contribute to the error rate as long as it is predicted incorrectly in either of the weaker learners $\{h_t^{k_{V1}}, h_t^{k_{V2}}\}$ got in step 3.

According to step 6, in every iteration, we calculate the percentage of the samples predicted same by two final weaker learners $\{h_t^{V1}, h_t^{V2}\}$ over the combined data set $X \cup S_t$, where S_t is the source data set finally chosen by us and $\{h_t^{V1}, h_t^{V2}\}$ are its weaker learners. Next, in order to indicate the characteristics of multi-view transfer learning in adaboost more deeply, we design step 7 to step 13 which are similar to the MV-TLAdaboost.

Algorithm MsTL-MvAdaboost

```
Input:
I. Source data sets: $S_1, \cdots, S_m$, where $n_{S_k}$ indicates the size of the source data set
   $S_k$ and $n_S = \sum_{k=1}^{m} n_{S_k}$.
II. Target data set: $X = (x_1^X, y_1^X), \cdots, (x_{n_X}^X, y_{n_X}^X)$, where $n_X$ indicates the size of
   the target data set $X$.
III. Initialize the weight vector $(\mathbf{w}^{S_1}, \cdots, \mathbf{w}^{S_m}, \mathbf{w}^X)$, where $\mathbf{w}^{S_k} = (w_1^{S_k}, \cdots, w_{n_{S_k}}^{S_k})$
   and $\mathbf{w}^X = (w_1^X, \cdots, w_{n_X}^X)$ to the desired distribution.
IV. Integer $T$ specifying number of iterations.

Divide all of the features into two views: $V1$ and $V2$
For $t = 1, 2, \cdots, T$
1. Normalize to 1 the weight vector: $(\mathbf{w}^{S_1}, \cdots, \mathbf{w}^{S_m}, \mathbf{w}^X)$
2. For $k = 1, 2, \cdots, m$
3.     Get back two weaker learners from two views: $\{h_t^{kV1}, h_t^{kV2}\} : X, S_k \rightarrow \{0, 1\}$
       over the combined data set $X \cup S_k$, weighted according to $(\mathbf{w}^X, \mathbf{w}^{S_k})$
4.     Calculate the error rate about the target data set $X$ by $\{h_t^{kV1}, h_t^{kV2}\}$ :
       $\varepsilon_t^k = \sum_{i=1}^{n_X} w_i^X \{max\{|h_t^{kV1}(x_i^X) - y_i^X|, |h_t^{kV2}(x_i^X) - y_i^X|\}\}$
       Note that, $\varepsilon_t^k$ is required to be less than 1/2
   End of For
5. Find the source data set $S_k$ and its weaker learners $\{h_t^{kV1}, h_t^{kV2}\}$ which contains
    the minimum error rate $\varepsilon_t^k$ and define:
    $\begin{cases} \{h_t^{V1}, h_t^{V2}\} = \{h_t^{kV1}, h_t^{kV2}\} \\ \varepsilon_t = \varepsilon_t^k \\ S_t = S_k \end{cases}$
6. Calculate the percentage of the samples predicted same by two final weaker
   learners $\{h_t^{V1}, h_t^{V2}\}$ over $X \cup S_t$ :
   $agree_t = 1 - \frac{\sum_{i=1}^{n_X} |h_t^{V1}(x_i^X) - h_t^{V2}(x_i^X)| + \sum_{i=1}^{n_{S_t}} |h_t^{V1}(x_i^{S_t}) - h_t^{V2}(x_i^{S_t})|}{n_X + n_{S_t}}$
7. Set $\epsilon_t = \varepsilon_t \, agree_t$
8. Set $\beta_t = \frac{\epsilon_t}{(1 - \epsilon_t)}$
9. Set $\beta = 1/(1 + \sqrt{2 \ln \frac{n_S}{T}})$
10. Calculate the distribution of $Xonly : R^t = \frac{w_i^X}{\sum_{j=1}^{n_X} w_j^X} \; for \; i = 1, \cdots, n_X$
11. Calculate the accuracy rate of $X$ with $\{h_t^{V1}, h_t^{V2}\}$ under the distribution
    $R^t = \{r_1^t, \cdots, r_{n_X}^t\}$ :
    $Acc_t^X = \sigma_t = \sum_{i=1}^{n_X} r_i^t \{1 - max\{|h_t^{V1}(x_i^X) - y_i^X|, |h_t^{V2}(x_i^X) - y_i^X|\}\}$
12. Calculate the overall accuracy rate of $X$ with $\{h_t^{V1}, h_t^{V2}\}$ under the general
    distribution $\mathbf{w}^X$ :
    $Acc_t^O = 1 - \varepsilon_t$
13. Set $\eta_t = \frac{Acc_t^O}{Acc_t^X}$
14. Set
    $\xi = max\{|h_t^{V1}(x_i^{S_k}) - y_i^{S_k}|, |h_t^{V2}(x_i^{S_k}) - y_i^{S_k}|\}$
    $\delta = max\{|h_t^{kV1}(x_i^{S_k}) - y_i^{S_k}|, |h_t^{kV2}(x_i^{S_k}) - y_i^{S_k}|\}$
    Update the weight vector:
    $\begin{cases} w_i^X = w_i^X (\beta_t \eta_t)^{-max\{|h_t^{V1}(x_i^X) - y_i^X|, |h_t^{V2}(x_i^X) - y_i^X|\}} \\ w_i^{S_k} = w_i^{S_k} \beta^{min\{\xi, \delta\}} \end{cases}$
End of For
Output the hypothesis:
$h_f(x) = \begin{cases} 1, & \text{if} \quad \sum_{t=1}^{T} \sum_{i=1}^{2} h_t^{Vi}(x) \geq T \\ 0, & \text{otherwise} \end{cases}$
```

Following this, in step 14, we use two different formulas to update the weight of samples from the target data set X and the source data set S_k. Details can be seen in (2).

$$\begin{cases} w_i^X = w_i^X (\beta_t \eta_t)^{-max\{|h_t^{V1}(x_i^X)-y_i^X|,|h_t^{V2}(x_i^X)-y_i^X|\}} \\ w_i^{S_k} = w_i^{S_k} \beta^{min\{\xi,\delta\}} \end{cases} \tag{2}$$

Two important points need to be illustrated here. On one hand, the target data set X is the core of our research and it is more representative than every source data set S_k. Thus, we want to make it prominent so that we increase the weight of the wrongly classified samples and keep the weight of the correctly ones.

On the other hand, not all the samples of source data sets are suitable for the target data set, so (2) provides a framework for automatically discovering which part of samples are specific for the source data sets only, which part may be more common between the target data set and the source data sets, and provides a way to distinguish these samples. At present, let us illustrate the effect of two parameters ξ and δ at first. We can see from step 2 to step 4 that, in every iteration, we will come to m pairs of weaker learners $\{h_t^{k_{V1}}, h_t^{k_{V2}}\}$ under their corresponding combined data sets $X \cup S_k$ and regard the one containing the minimum error rate as the final weaker learners $\{h_t^{V1}, h_t^{V2}\}$. However, different source data sets may contain different knowledge which means that $\{h_t^{V1}, h_t^{V2}\}$ may not be suitable to make a classification for all the source data sets. Therefore, for every sample in the source data set S_k, we design ξ to judge whether it is classified correctly by the final weaker learners $\{h_t^{V1}, h_t^{V2}\}$ and δ to judge whether it is classified correctly by its own weaker learners $\{h_t^{k_{V1}}, h_t^{k_{V2}}\}$. We regard the examples misclassified by both pairs of weaker learners $\{h_t^{V1}, h_t^{V2}\}$ and $\{h_t^{k_{V1}}, h_t^{k_{V2}}\}$ as the one which is not suitable to be transferred to the target data set and decrease its weight. On the contrary, if one sample can be classified correctly by either $\{h_t^{V1}, h_t^{V2}\}$ or $\{h_t^{k_{V1}}, h_t^{k_{V2}}\}$, we regard it as the one which is suitable for the target data set and keep its weight.

Finally, in the output step, we use the approach which is the same as MV-TLAdaboost to generate the final learner h_f .

4 Experiments and Results

In this part, in order to evaluate MsTL-MvAdaboost, we supply two source tasks simultaneously to help learn the target task. In all the experiments, we set the parameter View=2, which illustrates the number of views to be divided in the target task and source tasks.

Now we conduct experiments on several real data sets from the UCI repository. Above all, it is essential for us to illustrate that all the data sets used here are transformed into binary-classes problems of classification. Then, due to the characteristics of different data sets, we will use diverse ways to generate the target task and source tasks with different distributions to reach four sets of experiments on three real data sets.

On one hand, data sets $\frac{\text{Segmentation}}{\text{path:cement}}$ and $\frac{\text{Digit}}{5:8}$ are multi-classes problems of classification. As a result, we divide them into several binary-classes sub data sets by their labels to form the target task and source tasks. Details can be seen in Table 1.

Table 1. Summary of real data sets

Real data set	$\frac{\text{Segmentation}}{\text{path:cement}}$	$\frac{\text{Digit}}{5:8}$
Number of examples	1980	3361
Target training set	$\frac{550}{\text{path:cement}}$	$\frac{556}{5:8}$
Target testing set	$\frac{110}{\text{path:cement}}$	$\frac{556}{5:8}$
Size of the source task A	$\frac{660}{\text{sky:window}}$	$\frac{1115}{6:2}$
Size of the source task B	$\frac{660}{\text{grass:foliage}}$	$\frac{1134}{3:9}$
Dimensions	19	64
Number of classes	6	6

On the other hand, data sets $\frac{\text{Digit}}{3:8}$ and Landsat Satellite are binary-classes problems of classification. As a result, we divide them into several sub data sets by one special feature respectively to form the target task and source tasks. Details can be seen in Table 2.

Table 2. Summary of real data sets

Real data set	$\frac{\text{Digit}}{3:8}$	Landsat satellite
Number of examples	1126	2866
Target training set	154	504
Target testing set	154	504
Size of the source task A	330	815
Size of the source task B	488	1043
Dimensions	64	36
Number of classes	2	2

For every data set above, we set one special rule to divide it into the target task and source tasks with different distributions.

Segmentation is one seven-classes data set. We make use of all the data with label *path* and *cement* to generate the target task, the data with label *sky* and *window* to generate the source task A and the data with label *grass* and $foliage$ to generate the source task B.

Handwritten Digit is one ten-classes data set and we will use two different ways of generating the target task and source tasks to run the experiments. Firstly, according to the data set $\frac{\text{Digit}}{5:8}$ in Table 1, we make use of all the data with label 5 and 8 to generate the target task, the data with label 6 and 2 to generate the source task A, the data with label 3 and 9 to generate the source task B. Then, in Table 2, we get all the data with label 3 and 8 to come into one binary-classes data set to run another set of experiments, $\frac{\text{Digit}}{3:8}$. Now we divide it into the target task and source tasks on the basis of the value of *dimension six*. All the data according with the rule *dimension six* ≥ 10 belong to the target task, $5 \leq$ *dimension six* < 10 belong to the source task A and *dimension six* < 5 for the source task B.

In **Landsat satellite** data set, because the data of different classes are not balanced, we select all the samples with label *grey soil* and *very damp grey soil* to create one binary-classes data set. Furthermore, to divide this data set into the target task and source tasks, we set them by the *spectral value* of *dimension two*. The target task consists of all the data following the rule *dimension two* $>$ 100 while the source task A consists of all the data following the rule $80 \leq$ *dimension two* ≤ 100 and *dimension two* < 80 for the source task B.

Finally, it needs to notice that, in the experiments about MsTL-MvAdaboost and MV-TLAdaboost, due to the reason that we divide the dimensions about every data set into two views in half randomly, we run the experiments for ten times and get the mean of them as the final scores. Certainly, standard deviation (Std) will be calculated synchronously. Table 3 gives the classification error rates (Mean$_{\pm\text{Std}}$).

Table 3. Classification error rates

	$\frac{\text{Segmentation}}{\text{path:cement}}$	$\frac{\text{Digit}}{5:8}$	$\frac{\text{Digit}}{3:8}$	Landsat Satellite
Adaboost	0.0273	0.0162	0.0260	0.0258
$\frac{\text{MV-TLAda}}{\text{Source task A}}$	$0.0218_{\pm 0.0047}$	$0.0137_{\pm 0.0021}$	$0.0143_{\pm 0.0060}$	$0.0214_{\pm 0.0023}$
$\frac{\text{MV-TLAda}}{\text{Source task B}}$	$0.0227_{\pm 0.0048}$	$0.0153_{\pm 0.0019}$	$0.0176_{\pm 0.0081}$	$0.0226_{\pm 0.0028}$
MsTL-MvAda	$0.0164_{\pm 0.0038}$	$0.0104_{\pm 0.0027}$	$0.0078_{\pm 0.0080}$	$0.0185_{\pm 0.0021}$

Table 3 indicates clearly that our proposed algorithm, MsTL-MvAdaboost, comes to the best outcome in every data set.

5 Conclusion and Future Work

In this paper, we propose the algorithm, Multi-source Transfer Learning with Multi-view Adaboost (MsTL-MvAdaboost) to improve the effectiveness of transfer learning and prevent the influence of negative transfer by combining the characteristics of multi-source learning and the multi-view adaboost algorithm. After the detailed description of our algorithm, we run multiple experiments on real data sets to show its usefulness and effectiveness. In the future, we believe it can be an interesting challenge to extend the proposed MsTL-MvAdaboost algorithm to coping with more than two views and even with view learning simultaneously [8].

Acknowledgment. This work is supported by the Fundamental Research Funds for the Central Universities.

References

1. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22(10), 1345–1359 (2009)
2. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Boosting for transfer learning. In: Proceedings of the 24th International Conference on Machine Learning, pp. 193–200 (2007)
3. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Science 55, 119–139 (1997)
4. Xu, Z., Sun, S.: An algorithm on multi-view adaboost. In: Proceedings of 17th International Conference on Neural Information Processing, pp. 355–362 (2010)
5. Perkins, D.N., Salomon, G.: Transfer of learning. The Journal of International Encyclopedia of Education 2, 10 (1992)
6. Yao, Y., Doretto, G.: Boosting for transfer learning with multiple sources. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1855-1862 (2010)
7. Xu, Z., Sun, S.: Multi-view Transfer Learning with Adaboost. In: Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence, pp. 399–402 (2011)
8. Sun, S., Jin, F., Tu, W.: View Construction for Multi-view Semi-supervised Learning. In: Liu, D., Zhang, H., Polycarpou, M., Alippi, C., He, H. (eds.) ISNN 2011, Part I. LNCS, vol. 6675, pp. 595–601. Springer, Heidelberg (2011)

Semi-supervised Multitask Learning via Self-training and Maximum Entropy Discrimination

Guoqing Chao and Shiliang Sun

Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, P.R. China
guoqingchao10@gmail.com, slsun@cs.ecnu.edu.cn

Abstract. Maximum entropy discrimination (MED) is already shown to be effective for discriminative classification and regression, and can be applied to multitask learning (MTL) with some further assumptions. Self-training is a commonly used technique for semi-supervised learning. In order to integrate the merits offered by semi-supervised learning and MTL, this paper presents semi-supervised MTL via self-training and MED. We select the suitable measure metric and identify how to use unlabeled data. Experimental results on two UCI data sets demonstrate that our method yields better performance than semi-supervised single-task learning (STL) and supervised MTL.

Keywords: Semi-supervised learning, Multitask learning, Self-training, Maximum Entropy Discrimination.

1 Introduction

Multitask learning (MTL) has been proven to be an effective machine learning method to take advantage of the information contained in related tasks to improve the generalization performance. In general we consider many real-world problems in the single-task learning (STL) framework, but unavoidably ignore the useful information between these related tasks. Undoubtedly, MTL makes full use of these information by using them as an inductive bias, and utilizes a shared representation in implementation. Maximum entropy discrimination (MED), which was first presented in [1], can be conveniently used for MTL with some assumptions [2].

Supervised learning is a comparatively mature technique for classification with large number of labeled data to represent a sufficient sample from the true labeling function. However, in many real-world scenarios, acquiring labeled data is difficult owning to expensive human power and time consuming. Conversely, unlabeled data is easy to obtain, and much research indicates that unlabeled data plays a positive role in improving classification performance. Semi-supervised learning can learn a better classifier with both labeled data and unlabeled data, and self-training is one of the most commonly used techniques for semi-supervised learning.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 340–347, 2012.

In this paper, the motivation to present semi-supervised MTL via self-training and MED is to integrate the benefits provided by MTL and semi-supervised learning to make full use of the useful information from related tasks and the data distribution information from unlabeled data. There have been large amount of research on MTL such as [3,4], and semi-supervised learning also attracts a lot of attentions, including [5,6]. But to the best of our knowledge, there is few research on combination of the two techniques, except that Liu et al. [7] did similar work in this area.

The rest of the paper is organized as follows. Section 2 briefly reviews some related work on self-training and MTL via MED. Section 3 describes our proposed method semi-supervised MTL via self-training and MED. Section 4 demonstrates the experiments on two UCI data sets. Finally, we come to the conclusions and point out future work in Section 5.

2 Related Work

Self-training is classified to be a semi-supervised learning technique. The learning procedure is as follows: firstly, train a classifier with the small amount of labeled data; secondly, predict the labels of the large amount of unlabeled data, and add the most confident unlabeled data with the predicted labels to the training set; at last, train a new classifier with the new training set. The above three steps are repeated until prefixed iterations or certain stop conditions are satisfied.

The most significant characteristic of self-training is using its own predictions to teach itself. Therefore, this technique is easy to use for semi-supervised learning without complicated additional conditions. There have been large number of related work including [8,9], all of which obtain high performance improvements.

2.1 Multitask Learning via MED

This section refers from [1,2]. MTL via MED is mainly working as follows: MED firstly constructs a posterior that is as close as possible to the prior in terms of Kullback-Leibler Divergence, and then introduce a shared variable s in the likelihood function for MTL.

Given a collection of data sets $D = \{D_1, ..., D_M\}$ including $m = 1...M$ tasks. Each task has its training set D_m of $t = 1...T_m$ input output pairs $(x_{m,t}, y_{m,t})$, $x_{m,t} \in \mathbb{R}^D$ and $y_{m,t} = \{\pm 1\}$, task-specific model parameter Θ_m is corresponding to its data set D_m for $m = 1...M$. MED can be formulated as follows:

$$\begin{cases} \min_{p(\Theta|D)} \mathrm{KL}(p(\Theta|D) \parallel p(\Theta) \\ s.t. \int \log\Big(\dfrac{p(y_{m,t}|x_{m,t},\Theta_m)}{p(y|x_{m,t},\Theta_m)}\Big) p(\Theta|D) d\Theta \geqslant \gamma \\ \qquad \forall y \neq y_{m,t}, m, t. \end{cases} \tag{1}$$

The following posterior can be obtained according to the theorem in [1].

$$p(\Theta|D) = \frac{1}{Z(\lambda)} P(\Theta) \prod_{m=1}^{M} \prod_{t=1}^{T_m} \prod_{y \neq y_{m,t}} \Big(\frac{p(y_{m,t}|x_{m,t}, \Theta_m)}{p(y|x_{m,t}, \Theta_m)} \Big)^{\lambda_{m,t}} \exp(-\gamma \lambda_{m,t}). \quad (2)$$

The corresponding dual form is

$$\begin{aligned} \max\nolimits_{\lambda \geqslant 0} - \log \int p(\Theta) \prod_{m=1}^{M} \prod_{t=1}^{T_m} \prod_{y \neq y_{m,t}} \\ \Big(\frac{p(y_{m,t}|x_{m,t}, \Theta_m)}{p(y|x_{m,t}, \Theta_m)} \Big)^{\lambda_{m,t}} \exp(-\gamma \lambda_{m,t}) d\Theta. \end{aligned} \quad (3)$$

Now, a shared variable s is introduced for MTL, making the likelihood function changes into:

$$p(y|x, \Theta_m, s) \propto \exp\Big(\frac{y}{2} (\sum_{d=1}^{D} s(d) x(d) \theta_m(d) + b_m) \Big). \quad (4)$$

Here, s is a binary vector to choose corresponding entry of x uniformly for all tasks, and its prior is assumed as follows:

$$p(s) = \prod_{d=1}^{D} \rho^{s(d)} (1 - \rho)^{1 - s(d)}. \quad (5)$$

Finally, formula (3) transforms to the following form:

$$\begin{cases} \max_\lambda \sum_{m=1}^{M} \sum_{t=1}^{T_m} \gamma \lambda_{m,t} - \sum_{d=1}^{D} \log(\alpha + e^{\frac{1}{2} \sum_{m=1}^{M} (\sum_{t=1}^{T_m} \lambda_{m,t} y_{m,t} x_{m,t}(d))^2}) \\ \qquad + D\log(\alpha + 1) \\ s.t. \;\; 0 \leq \lambda_{m,t} \leq C \; \forall m, t \\ \qquad \sum_{t=1}^{T_m} y_{m,t} \lambda_{m,t} = 0 \; \forall m. \end{cases} \quad (6)$$

When the λ setting is obtained, the following formula is used to predict the label of a new query.

$$\hat{y} = \underset{y}{\mathrm{argmax}} \mathrm{E}_{p(\Theta|D)} [\mathrm{log} p(y|x, \Theta_m, s)] \quad (7)$$

The algorithm presented in [2] is shown in Table 1.

3 Semi-supervised MTL via Self-training and MED

Semi-supervised MTL via self-training and MED introduces self-training into MTL via MED. For self-training, there are two problems to be solved in Section 3.1. One is selecting what metric to measure the confidence of the predicted label of the unlabeled data, the other is how to add two classes of unlabeled data into the training set. Section 3.2 illustrates the algorithm of the new semi-supervised MTL method.

Table 1. Algorithm 1: Multitask MED learning

0	Input data set $D, C > 0, \alpha \geq 0, 0 < \varpi < 1$ and kernels k_d for $d = 1, ..., D$.
1	Initialize Lagrange multipliers to zero $\lambda = \mathsf{0}$.
2	Store $\tilde{\lambda} = \lambda$.
3	For $m = 1, ..., M$ do:
3a	Set $g_d = \alpha \exp\left(-\frac{1}{2}\sum_{m=1}^{M}\sum_{t=1}^{T_m}\lambda_{m,t}\lambda_{m,\tau}y_{m,t}y_{m,\tau}k_d(x_{m,t}x_{m,\tau)})\right)$ for all d. Set $G_d = \frac{\tanh(\frac{1}{2}\log(g_d))}{2\log(g_d)}$ for all d. Set $\hat{s}(d) = \frac{1}{1+g_d}$ for all d. Set $\hat{y}_{m,t}(d) = \sum_{t=1}^{T_m}\lambda_{m,\tau}y_{m,\tau}k_d(x_{m,t}, x_{m,\tau})$. for all t and d.
3b	Update each of the λ_m vectors with the SVM QP: $\max_{\lambda_m}\sum_{t=1}^{T_m}\lambda_{m,t} - \sum_{t=1}^{T_m}\lambda_{m,t}y_{m,t}\sum_{d=1}^{D}$ $\hat{s}(d)\hat{y}_{m,t}(d) + \sum_{t=1}^{T_m}\sum_{\tau=1}^{T_m}\lambda_{m,t}\tilde{\lambda}_{m,\tau}y_{m,t}y_{m,\tau}$ $\sum_{d=1}^{D}\left(G_d\hat{y}_{m,t}(d)\hat{y}_{m,\tau}(d) + k_d(x_{m,t}, x_{m,\tau})\right)$ $-\frac{1}{2}\sum_{t=1}^{T_m}\sum_{\tau=1}^{T_m}\lambda_{m,t}\lambda_{m,\tau}y_{m,t}y_{m,\tau}\sum_{d=1}^{D}$ $\left(G_d\hat{y}_{m,t}(d)\hat{y}_{m,\tau}(d) + k_d(x_{m,t}, x_{m,\tau})\right)$ $s.t.\ 0 \leq \lambda_{m,t} \leq C \quad \forall t = 1, ..., T_m$ and $\sum_{t=1}^{T_m}y_{m,t}\lambda_{m,t} = 0$.
4	If $\left\Vert\lambda - \tilde{\lambda}\right\Vert > \varpi \Vert\lambda\Vert$ go to 2.
5	Output: $\hat{\mathsf{s}}$ and λ.

3.1 Selected Metric and How to Add Unlabeled Data

We choose the predicted real number y as selected metric, since if the y predicted by MTL via MED is larger, its label is more possible to be positive, and otherwise its label tends to be negative, which nicely meets the requirements of the selected metric.

As to the latter problem, we make the following processing: calculate the ratio of the two class data in training set; and then use the ratio to be that of predicted two class data which need to be added into training set. It's necessary to make such a treatment, because the distribution of two class data is not balanced. Assume that there are 95 positive data and 5 negative data in the training set, and there are 1000 unlabeled data. In general, the distribution of the training set and testing set are similar. Therefore, there should be about 950 positive data and 50 negative data for unlabeled data. If we add the same number of two classes data into the training set, for instance, 50, perhaps we cannot get any performance gain. But obviously, we don't make full use of the unlabeled data. If following our proposed way, choose 95 positive data and 5 negative data to be added each iteration, and maybe we can repeat near to 10 iterations with performance increase.

3.2 Algorithm

In order to implement the semi-supervised MTL via self-training and MED, we introduce self-training to Algorithm 1 to obtain Algorithm 2. The new algorithm is shown in Table 2.

Table 2. Algorithm 2: Semi-supervised MTL via self-training and MED

0	Given: L: labeled data set; U: unlabeled set (for predicting to add); T: unlabeled set (for validation and testing).
1	Utilize algorithm 1 to train on L and predict on U to choose the most confident unlabeled data to add into L (Noted that the way to choose confident unlabeled data is according to the way we mentioned former).
2	Repeat step1 for some iterations to stop.
3	Output the accuracy of classification.

4 Experiments

In this section, we compare semi-supervised MTL via self-training and MED against: (1) supervised MTL, (2) semi-supervised STL. MTL refers to learning the classifiers with MTL via MED, but STL means learning all classifiers independently. We don't compare with supervised STL, that's because large amount of research have verified that supervised MTL and semi-supervised STL perform better than supervised STL.

To show their performance, we utilize the accuracy of classification to be the performance measure. In addition, we will determine the value of C for STL and the values of C and α by cross-validation on held out data and then test on an unseen testing set. The following part will illustrate the experiments on two UCI data sets.[1]

Dermatology. The dermatology data set is one task of 6 classes which can be converted into 6 binary classification tasks. There are totally 366 instances and each instance has 34-dimensional feature. For two semi-supervised learning methods, we will train on various numbers of examples (from 5 to 150) for each task, and employ the following 100 examples as unlabeled data, and the remaining examples are split in half for cross-validation and testing. Correspondingly, supervised MTL will be done the same processing except using the unlabeled examples, the experiment result is shown in Fig. 1.

[1] Data available at http://archive.ics.uci.edu/ml/

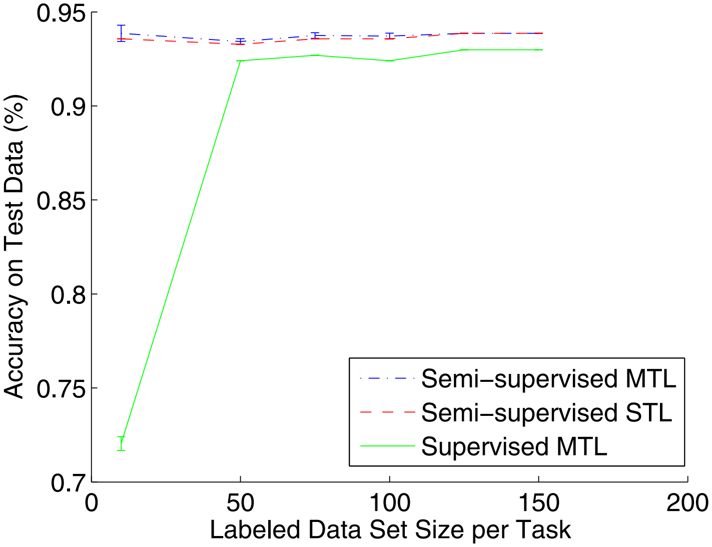


Fig. 1. The performance comparison of three learning methods on Dermatology data set

Glass. The glass data set consists of 6 classes task which will also be converted into 6 binary classification tasks, which has 214 instances whose feature is 10-dimensional. We will do similar processing with dermatology. The number of training set will be from 10 to 80 for each task, and the unlabeled data will be the following (from 150 to 80) examples. The remaining examples will be split in half for cross-validation and testing. The experiment result is illustrated in Fig. 2.

From Fig. 1 and Fig. 2, we can clearly find that the performance of semi-supervised MTL via self-training and MED is higher than that of two other methods when the labeled data points are less than 50 and 30 respectively. After that, semi-supervised STL catches up, but our method still outperforms supervised MTL. When the labeled data amounts to certain number, semi-supervised STL shows the same performance with semi-supervised MTL via self-training and MED, that may be attributed to the doubly enhanced information offered by related tasks and unlabeled data, which significantly augment the information in the labeled data.

The experimental results shows the new method is superior, that's because we make full use of the information from related works and unlabeled data sets. But it cannot guarantee its superior in any case. We argue that the predicted label of unlabeled data must be highly confident and right, and the shared structure exists in multiple tasks. In fact, we will further study the underlying cause in future work.

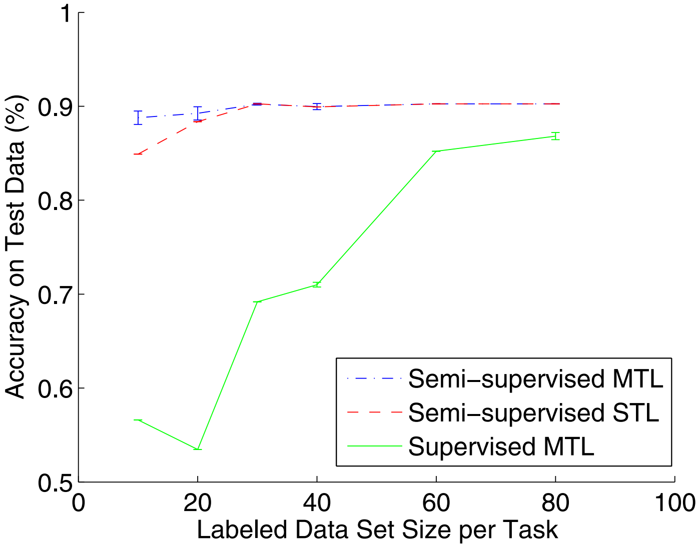


Fig. 2. The performance comparison of three learning methods on Glass data set

5 Conclusion and Future Work

This paper applies self-training to MTL via MED to obtain one semi-supervised MTL method. We make necessary modification for this extension. And the experimental results on two UCI data sets illustrate that this method is superior to other two ones.

Concerning future work, we can consider combining other semi-supervised learning method with other MTL technique to get another new semi-supervised MTL method. Moreover, it may be interesting to integrate active learning [10,11] with MTL via MED if the label of the selected informative data can be available. Analyzing in which cases the self-training will benefit the MED and in which cases it is not still promising is also meaningful.

Acknowledgment. This work is supported in part by the National Natural Science Foundation of China under Project 61075005, and the Fundamental Research Funds for the Central Universities.

References

1. Jaakkola, T., Meila, M., Jebara, T.: Maximum Entropy Discrimination. In: Proceedings of Advances in Neural Information Processing Systems (1999)
2. Jebara, T.: Multitask Sparsity via Maximum Entropy Discrimination. Journal of Machine Learning Research, 75–110 (2011)

3. Caruana, R.: Multitask Learning. Machine Learning, 41–75 (1999)
4. Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multi-task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 35–63 (2007)
5. Chapelle, O., Schlkopf, B., Zien, A.: Semi-supervised Learning. MIT Press (2006)
6. Sun, S.: Multi-view Laplacian Support Vector Machines. In: Tang, J., King, I., Chen, L., Wang, J. (eds.) ADMA 2011, Part II. LNCS, vol. 7121, pp. 209–222. Springer, Heidelberg (2011)
7. Liu, Q., Liao, X., Carin, L.: Semi-supervised Multi-task Learning. In: Proceedings of Advances in Neural Information Processing Systems (2007)
8. McClosky, D., Charniak, E., Johnson, M.: Effective Self-training for Parsing. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pp. 152–159 (2006)
9. Rosenberg, C., Hebert, M., Schneiderman, H.: Semi-supervised Self-training of Object Detection Models. In: Proceedings of Seventh IEEE Workshop on Applications of Computer Vision, pp. 29–36 (2005)
10. Sun, S.: Active Learning with Extremely Sparse Labeled Examples. Neurocomputing, 2980–2988 (2010)
11. Zhang, Q., Sun, S.: Multiple-view Multiple-learner Active Learning. Pattern Recognition, 3113–3119 (2010)

Tracking Property of UMDA in Dynamic Environment by Landscape Framework

Ran Long[1], Liangqi Gong[1], Bo Yuan[1], Ping Ao[2], and Qingsheng Ren[1]

[1] Department of Computer Science and Engineering,
Shanghai Jiao Tong University,
200240 Shanghai, China
{longran1989,lqgong.sjtu}@gmail.com, {yuanbo,ren-qs}@cs.sjtu.edu.cn
[2] Shanghai Center for Systems Biomedicine,
Shanghai Jiao Tong University, Shanghai, China
aoping@sjtu.edu.cn

Abstract. In this paper, the landscape framework is used to analysis the tracking performance of univariate marginal distribution algorithm (UMDA) in dynamic environment. A set of stochastic differential equations (SDEs) is used to describe the evolutionary dynamics of the algorithm. The corresponding potential function is constructed from these SDEs. Dynamic mean first passage time, which is a new concept, is defined as the time it takes from an optimum to another in a dynamic environment. This concept can be used to measure the tracking property of the algorithm.

Keywords: UMDA, Dynamic environment, Potential function, Dynamic mean first passage time.

1 Introduction

Now Evolutionary Algorithms(EAs) have been a popular means for solving optimization problems in dynamic environment [1]. All the solutions could be mainly separated into four categories: changing the EA strategy with the variation of environment [2], maintaining diversity throughout the run [3], importing some memory-based approaches [4] and multi-population approaching [5, 6].

However, there is not much work coming from a theoretical point of view. In the very recent past, much more researchers started to investigate in a theoretical way. First, through some investigation of a (1+1) EA on the changing environment, the transition probabilities were given for the first time [7]. And then, Droste gave three papers [8–10], in which the first passage time was proposed.

In this paper, we apply the adaptive landscape framework to UMDA in a dynamic environment. Adaptive landscape was first raised by Wright [11]. It has prevailed well in population genetics in particular and biology. In 2002, the idea of using this framework to study EAs was applied by [12], which showed the evolution occurred by gradient ascent in a landscape. [11] showed that the population could transverse across a saddle configuration from a stable point to

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 348–355, 2012.

another one with better fitness. This phenomenon would never happen in gradient ascent environment. Only noise can help the system hop from one optimum to another.

In Section 2, the UMDA is modeled by a set of stochastic differential equations (SDEs). The corresponding potential function is constructed from the SDEs. The function gives a global prospect of the algorithm, which helps us recognize the evolutionary behavior clearly. In Section 3, we calculate the dynamic mean first passage time by an approximation method. This new concept is used to describe how long it will take from an optimum to another in average. It is a new way to measure the tracking property of UMDA in dynamic environment.

2 Methodology – The Landscape framework

2.1 SDE Model of UMDA

Algorithm.1 shows the pseudo code for a general UMDA algorithm.

Algorithm 1. Pseudo code for a general UMDA algorithm

```
INITIALIZE probability vector $p(\mathrm{x}, 0)$
**repeat**
    GENERATE N individuals according to $p(\mathrm{x}, t)$
    EVALUATE them with respect to $f(\mathrm{x})$
    SELECT M $\leq$ N best individuals to calculate the frequencies $p^s(x_i, t)$
    UPDATE $p(\mathrm{x}, t)$: $p(\mathrm{x}, t+1) = \prod_{i=1}^{n} p^s(x_i, t)$
    $t = t + 1$
**until** TERMINATION CONDITION is satisfied
VARIABLE: $p(\mathrm{x}, t)$ is the probability vector at the $t$th step.
```

According to [12], the updating rule of UMDA can be described by the following ordinary differential equations (ODEs) under the assumption of infinite population and proportional selection:

$$\dot{p}_i = \frac{p_i(1-p_i)}{W} \times \frac{\partial W}{\partial p_i} \tag{1}$$

where $W = \sum p(\mathrm{x}, t) f(\mathrm{x})$. $\dot{p}_i = dp_i/dt, i = 1, 2, ..., n$, $p_i = p(x_i = 1, t)$. The evolution of p_i is deterministic in the probability space although UMDA itself is stochastic. The system approaches the optima/attractor with a gradient descent, which is determined by its initial condition [13, 12].

In [12], it is said, ”For difficult optimization problems, there exists a huge number of attractors, each with a corresponding attractor region. If the iteration starts at a point within the attractor region, it will converge to the corresponding attractor at the boundary. But if the iteration starts at points which lie at the boundary of two or more attractors, i.e. on the separatrix, the iteration will be confined to the separatrix. The deterministic system cannot decide for one of the attractors."

UMDA with a finite population does not have a sharp boundary between attractor regions. We model this behavior by introducing randomness:

$$\dot{p}_i = \frac{p_i(1-p_i)}{W} \times \frac{\partial W}{\partial p_i} + \zeta_i(\mathrm{p}, t) \tag{2}$$

where ζ_i is a Gaussian and white noise added in the algorithm, satisfies:

$$< \zeta_i >= 0 \tag{3}$$

$$< \zeta_i(\mathrm{p}, t), \zeta_j(\mathrm{p}, t') >= 2d_{ij}(\mathrm{p})\delta(t - t') \tag{4}$$

Here $\delta()$ is the dirac delta function, matrix $\mathbf{D} = (d_{ij})$ indicates the diffusion matrix. If we assume the diffusion does not have interactions in different dimensions, we could set the diffusion matrix as a diagonal matrix. In this paper, we set all the diagonal elements to be the same for simplicity:

$$\mathbf{D} = diag\{d, d, \ldots, d\} \tag{5}$$

The stochasticity added in the algorithm is also analogous to mutation in EAs. In [12], the authors add mutation into their algorithm. The algorithm is still be described by a set of ODEs, and that kind of mutation moves the stable points from the boundary into the interior. Similar results can be found in [17, 18]. But in our work, the stable points remain at the boundary of the search space whatever the level of the noise.

2.2 Potential Function

The potential function can be used to get a global description about the evolutionary behavior of the dynamical system. Let us consider the dynamical system described by a set of SDEs, as the component equation [19]:

$$\dot{x}_j = f_j(\mathrm{x}) + \zeta_j(\mathrm{x}, t) \tag{6}$$

The state variable forms an n dimensional vector $\mathrm{x}^T = (x_1, x_2, \ldots, x_n)$. $f_j(\mathrm{x})$ is the deterministic factor on the jth component, which includes both the effects from other components and itself. $\zeta_j(\mathrm{x}, t)$ is the random factor. For simplicity we assume that f_j is a smooth function explicitly independent of time t.

In [14, 15] the above equation is rewritten in the following form

$$[S(\mathrm{x}) + A(\mathrm{x})]\dot{\mathrm{x}} = -\nabla\Phi(\mathrm{x}) + \xi(\mathrm{x}, t) \tag{7}$$

Here ∇ is the gradient operator in the state variable space. The single-valued scalar function, $\Phi(\mathrm{x})$, plays a role as a potential energy in physical sciences. The semi-positive definite symmetric matrix $S(\mathrm{x})$ is dissipative and its dynamic effect is to decrease the potential $\Phi(\mathrm{x})$. It implies the system's ability to find the point with smallest potential value. The anti-symmetric matrix $A(\mathrm{x})$ is non-dissipative and it leads no change in potential. ξ is the stochastic force, by which the system can hop from one potential valley to another.

Based on the equivalence of equation (6) and (7), we can get[14, 15]

$$\Phi(\mathrm{x}) = -\int_C d\mathrm{x}' \cdot \left[G^{-1}(\mathrm{x}')f(\mathrm{x}')\right] \tag{8}$$

$$G(\mathrm{x}) = [S(\mathrm{x}) + A(\mathrm{x})]^{-1} \tag{9}$$

The end and initial points of the integration contour C are x_t and x_0.

3 Tracking Performance of UMDA in Dynamic Environment: A Case Study

3.1 Dynamic Wright's Fitness Function

The example we use is based on Wright's fitness function [11, 12].

$$\begin{aligned} maxh(\mathrm{x},t) = {} & \frac{13}{2}(x_1+x_2) + \frac{14}{2}(x_3+x_4) + (x_1+x_2-x_3-x_4)cos\omega t \\ & -4(x_1x_2+x_1x_3+x_1x_4+x_2x_3+x_2x_4+x_3x_4) \end{aligned} \tag{10}$$

where $\mathrm{x} = (x_1,x_2,x_3,x_4)^T$,$x_i \in \{0,1\}$. ω is angular velocity which controls the rate of function changing. Sometimes we consider the phase ωt directly.

The reasons we use this function are:

- There are at least two peaks (one local and one global).
- The peaks do not change the position, but the height changes with the time. The local optimum may become global, and vice versa. The global optimum moves from (1,1,0,0) to (0,0,1,1) when ωt changes from 0 to π.

Then, the average fitness function W is:

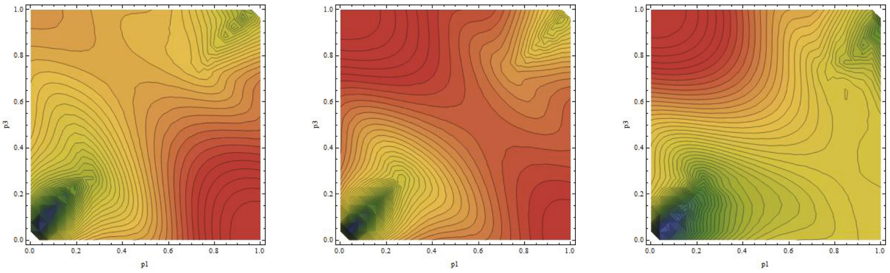
$$\begin{aligned} W(\mathrm{p},t) = {} & \frac{13}{2}(p_1+p_2) + \frac{14}{2}(p_3+p_4) + (p_1+p_2-p_3-p_4)cos\omega t \\ & -4(p_1p_2+p_1p_3+p_1p_4+p_2p_3+p_2p_4+p_3p_4) \end{aligned} \tag{11}$$

In order to visualize the potential landscape, we make a simplification. Those equations are symmetric in p_1,p_2 and p_3,p_4, so we set $p_1 = p_2$ and $p_3 = p_4$. The new average fitness function will be written as:

$$W\ \ (\mathrm{p},t) = 13p_1 + 14p_3 + 2(p_1-p_3)cos\omega t - 4(p_1^2 + 4p_1p_3 + p_3^2) \tag{12}$$

3.2 Potential Function

Fig. 1 shows the contour of potential function at different phase. From these pictures, we can see the stable points remain at the boundary of the search space. (1,0) is the global optimum when $\omega t = 0$ while (0,1) is the global when $\omega t = \pi$. But the saddle point changs with the time.



(a) $\omega t = 0$ (b) $\omega t = \pi/2$ (c) $\omega t = \pi$

Fig. 1. The Potential Function(red is more adaptive)

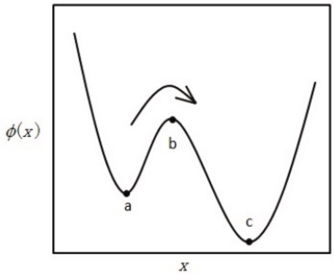


Fig. 2. Definition of Mean First Passage Time (Local optimum a, saddle point b, global optimum c)

3.3 Dynamic Mean First Passage Time

In the static environment, mean first passage time is used to show how long the system will take from an optimum to another in average [16]. In Fig. 2, a is local optimum and c is global optimum. With the help of noise, the system can hop from one optimum through saddle point b to another. The mean first passage time from a to b obeys the following relation:

$$T_{ab} \propto exp(-(\Phi_a - \Phi_b)) \tag{13}$$

This passage time is a quantitative measure of robustness of the system. With long passage time, the system stays at a certain optimum with a large probability. With short passage time, the system is unstable because it can hop from one optimum to another easily.

For a static problem, the passage time is a constant. But in the dynamic environment, things are different because the difference between the potential value will change with time. We define $\gamma(\omega)$, which satisfies the following equation, as dynamic mean first passage time:

$$\int_0^{\gamma(\omega)} \frac{dt}{M(\omega t)} = 1 \tag{14}$$

Table 1. The value of C

d	0.01	0.02	0.03	0.04	0.05
C	15.533	10.463	7.760	6.535	5.734

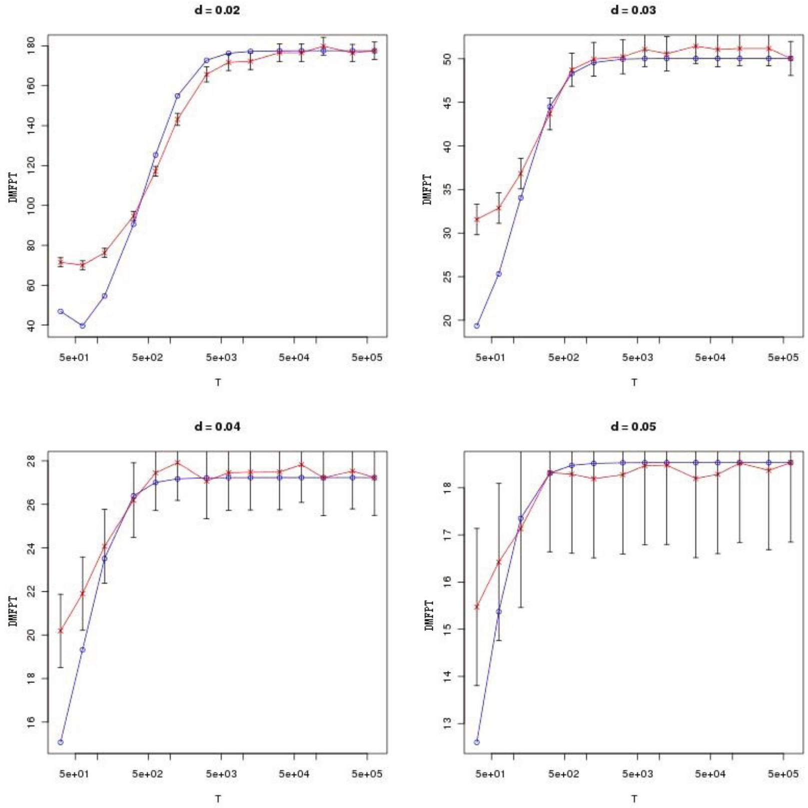


Fig. 3. The Dynamic Mean First Passage Time($T = 2\pi/\omega$)

Where $M(\omega t)$ is the stationary mean first passage time. It is obviously that the algorithm can not track the optima when $\gamma(\omega) > \frac{2\pi}{\omega}$.

In order to get $\gamma(\omega)$, we first give the following 1-order approximation of M:

$$M = Cexp(-(\Phi_a - \Phi_b)) \tag{15}$$

By numerical experiment, we get the value of C which is related with d (see Table 1). Then we can get $\gamma(\omega)$ for different d.

Another experiment is made for checking whether the 1-order approximation is suitable. In the Fig 3, the red one is got from UMDA directly and the blue one is obtained by the approximation method.

Table 2. The Dynamic Convergence Ratio

$d \backslash \omega$	2e-1	1e-1	5e-1	2e-2	1e-2	5e-3	2e-3	1e-3	5e-4	2e-4	1e-4
0.02				1.5475	1.7638	2.0236	3.3421	4.9984	8.2449	8.6491	8.8264
0.03			1.4559	1.6079	1.6394	2.3138	5.4471	7.4231	9.1487	9.2341	
0.04		1.2992	1.2925	1.4198	1.6651	3.4195	5.7035	7.4067	6.7602		
0.05	1.0150	1.1825	1.3062	1.2430	1.8636	3.0636	4.7764	4.7738			

We also calculate the dynamic convergence ratio dr:

$$\mathrm{dr} = \frac{\text{the number that the system is at the current global optimum}}{\text{the number that the system is at the current local optimum}} \tag{16}$$

The simulation of SDEs are simulated for 50 periods ($T = 2\pi/\omega$) and the results are shown in Table 2.

According to the results of Fig. 3 and Table. 2, we get the following facts:

- The convergence ratio is larger than 1 all the time.
- If the objective function changes slowly, small ω or great T, the dynamic mean first passage time is long and the convergence ratio is high.
- The relation between diffusion coefficient d and convergence ratio is complex. If d is small, the dynamic mean first passage time is long. The tracking speed is low and the convergence ratio is low. If d is large, the dynamic mean first passage time is short. The system can hop from local to global easily. But it also can hop from global to local easily. The convergence ratio is still low.

4 Conclusion

In this paper, we present the landscape framework to study the behavior of UMDA in dynamic environment. SDEs are used to model the algorithm, and the potential function is constructed to help revealing the entire search space. Dynamic mean first passage time is defined to describe the tracking property in the dynamic environment. This framework cannot handle all the cases now. Like [20], we can only analysis the situation that the original problem is polynomial. Further exploration is need for the visualization of landscape when dimensions are higher than 2. Our future work also focus on the theory analysis of other scenario, such as multi-population and multi-objective.

Acknowledgments. This work is supported in part by the China NSF Grants 61073087 and 91029738, The National Basic Research Program of China (973 Program) under Grant No.2007CB914700 and No.2010CB529200.

References

1. Jin, Y., Branke, J.: Evolutionary Optimization in Uncertain Environments: A Survey. IEEE Transaction on Evolutionary Computation 9(3), 303–317 (2005)

2. Cartwright, H., Tuson, A.: Genetic Algorithms and Flowshop Scheduling: Towards the Development of A Real-time Process Control System. Evolutionary Computing, 277–290 (1994)
3. Grefenstette, J.J.: Genetic Algorithms for Changing Environments. In: Manner, R., Manderick, B. (eds.) Parallel Problem Solving from Nature, pp. 137–144. Elsevier (1992)
4. Goldberg, D.E., Smith, R.E.: Nonstationary Function Optimization Using Genetic Algorithm with Dominance and Diploidy. In: Proc. of the 2nd Int. Conf. on Genetic Algorithms, pp. 59–68 (1987)
5. Branke, J., Kaußler, T., Thomas, K., Christian, S., Hartmut, S.: A Multi-population Approach to Dynamic Optimization Problems. In: Adaptive Computing in Design and Manufacturing, pp. 299–308. Springer (2000)
6. Oppacher, F., Wineberg, M.: The Shifting Balance Genetic Algorithm: Improving the GA in a Dynamic Environment. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 504–510 (1999)
7. Stanhope, S.A., Daida, J.M.: (1+ 1) Genetic Algorithm Fitness Dynamics in a Changing Environment. In: Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1851–1858. IEEE (1999)
8. Droste, S.: Analysis of the (1+ 1) EA for a Dynamically Changing Objective Function. HT014601767. University Dortmund (2001)
9. Droste, S.: Analysis of the (1+ 1) EA for a Dynamically Changing Onemax-variant. In: Proceedings of the 2002 Congress on Evolutionary Computation, pp. 55–60. IEEE (2002)
10. Droste, S.: Analysis of the (1+ 1) EA for a Dynamically Bitwise Changing OneMax. 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. 2724, p. 202. Springer, Heidelberg (2003)
11. Wright, S.: The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution. In: Proc. of the 6th Inter. Congress of Genetics, pp. 356–366 (1932)
12. Muhlenbein, H., Mahnig, T.: Evolutionary Computation and Wright's Equation. Theoretical Computer Science 287(1), 145–165 (2002)
13. Gonzalez, G., Lozano, J.A., Larranaga, P.: Mathematical Modeling of Discrete Estimation of Distribution Algorithms. In: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, pp. 147–163. Kluwer Academic (2002)
14. Yin, L., Ao, P.: Existence and Construction of Dynamical Potential in Nonequilibrium Processes without Detailed Balance. Journal of Physics A: Mathematical and General 39, 8593–8601 (2006)
15. Ao, P.: Potential in Stochastic Differential Equations: Novel Construction. Journal of Physics A: Mathematical and General 30, L25–L30 (2004)
16. van Kampen, N.G.: Stochastic Processes in Physics and Chemistry. North-Holland (2007)
17. Mahnig, T., Mulenbein, H.: Optimal Mutation Rate Using Bayesian Priors for Estimation of Distribution Algorithms. In: Steinhöfel, K. (ed.) SAGA 2001. LNCS, vol. 2264, pp. 460–463. Springer, Heidelberg (2001)
18. Hisashi, H.: The Effectiveness of Mutation Operation in the Case of Estimation of Distribution Algorithms. Biosystems 87, 243–251 (2007)
19. Gardiner, C.W.: Handbook of Stochastic Processes. Springer (1991)
20. Mahnig, T., Muhlenbein, H.: Mathematical Analysis of Optimization Methods Using Search Distributions. In: Proceedings of the 2000 Genetic and Evolutionary Computation Conference, pp. 205–208 (2000)

Computation of Joint Spectral Radius for Network Model Associated with Rank-One Matrix Set*

Jun Liu and Mingqing Xiao

Department of Mathematics,
Southern Illinois University, Carbondale, IL 62901-4408, USA
{jliu,mxiao}@math.siu.edu

Abstract. In the paper, we prove that any finite set of rank-one matrices has the finiteness property by making use of (invariant) extremal norm. An explicit formula for the computation of joint/generalized spectral radius of such type of matrix sets is derived. Several numerical examples from current literature are provided to illustrate our theoretical conclusion.

Keywords: joint/generalized spectral radius, finiteness property, irreducible, extremal norm, Barabanov norm.

1 Backgrounds

The joint spectral radius of a finite set of matrices plays an important role in many applications, such as wavelet theory [5, 9–11, 26], stability of switched linear systems [7, 8, 17, 29], and their references therein. Hence, its computation or approximation is of great interest in reality.

We consider the following network model governed by switched linear systems

$$x_{\ell+1} = A_{\omega_\ell} x_\ell, \qquad \ell \geq 0 \tag{1}$$

where $x_\ell \in \mathbb{R}^n$ with $n \geq 2$ be fixed, $A_{\omega_\ell} \in \mathbb{R}^{n\times n}$ with ω_ℓ taking values in a finite set $\mathcal{A} = \{1, \ldots, m\}$ with $m \geq 2$. Let Σ_κ be the set of all possible mappings

$$\omega \colon \mathbb{Z}_+ \to \mathcal{A},$$

where $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$ denotes the nonnegative integer set. Each element in Σ_κ is called a *switching sequence* of system (1). It is well-known that growth rate of the network dynamics is determined by the joint spectral radius of the matrix set $\{A_i : i \in \mathcal{A}\}$. Hence the computation of joint spectral radius becomes critical in order to identify the stability of the network dynamics.

Let $\|\cdot\|$ be any sub-multiplicative matrix norm and $\rho(A)$ be the spectral radius of a matrix A. Given a finite set $\mathcal{F} = \{A_1, A_2, \cdots, A_m\} \subset \mathbb{C}^{n\times n}$ of

* This work was supported by NSF 1021203 of the United States.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 356–363, 2012.

complex $n \times n$ matrices. We define the set $\mathcal{F}_k$ of all possible products of length $k \geq 1$ with factors from $\mathcal{F}$, i.e.,

$$\mathcal{F}_k = \{A_{i_1} A_{i_2} \cdots A_{i_k} : \quad 1 \leq i_j \leq m,\ j = 1, \ldots, k\}.$$

The joint spectral radius of $\mathcal{F}$ is defined by [28]

$$\hat{\rho}(\mathcal{F}) = \lim_{k \to \infty} \max_{A \in \mathcal{F}_k} \|A\|^{1/k},$$

and its generalized spectral radius by [9]

$$\bar{\rho}(\mathcal{F}) = \limsup_{k \to \infty} \max_{A \in \mathcal{F}_k} \rho(A)^{1/k}.$$

Since the equality $\hat{\rho}(\mathcal{F}) = \bar{\rho}(\mathcal{F})$ has been established for any finite set of matrices [1, 12], we designate an unified notation $\rho(\mathcal{F})(= \hat{\rho}(\mathcal{F}) = \bar{\rho}(\mathcal{F}))$ in the following. Another equivalent variational way of characterizing joint spectral radius is [28]

$$\rho(\mathcal{F}) = \inf_{\|\cdot\|} \max_{A \in \mathcal{F}} \|A\|, \tag{2}$$

where the infimum is taken over the set of all sub-multiplicative matrix norms. Whenever the infimum in (2) is attained (thus a minimum), the corresponding norm $\|\cdot\|_*$ will be called an extremal norm [30]. The definition (2) is somehow attractive in the sense that its estimation of $\rho(\mathcal{F})$ avoids the computation of long matrix products as long as $\|\cdot\|_*$ is efficiently computable. Straightforward algorithms [13, 21] for computing or approximating $\rho(\mathcal{F})$ mostly make use of the following three important inequalities

$$\max_{A \in \mathcal{F}_k} \rho(A)^{1/k} \leq \rho(\mathcal{F}) \leq \max_{A \in \mathcal{F}_k} \|A\|^{1/k} \tag{3}$$

for any $k \geq 1$. In general, however, such a brute-force approach is far from satisfactory and the highly slow convergence renders this estimation impractical to many problems, in particular, for those large-scale ones. In order to obtain better approximations within current computational capacity, many numerical methods were proposed during last decade as we categorize in the following.

The first approach is to try to construct the extremal norm $\|\cdot\|_*$ or at least approximate it when it exists. One necessary and sufficient condition for the existence of an extremal norm is the non-defectiveness of the corresponding normalized matrix family [14], which is generally not algorithmically decidable [4]. In [3], the minimization was restricted to the set of ellipsoid norms, which can be efficiently approximated by current convex optimization algorithms. This approach provides a theoretical precision estimation of $\rho(\mathcal{F})$ in limited applicable cases. In [14, 16, 15], the minimization was confined to the set of complex polytope norms. The successful construction of such a polytope norm is not always guaranteed, and it is more suitable to be used to verify the occurrence of the finiteness property of $\mathcal{F}$ [20], that is, to check the case when there is a positive integer t such that

$$\rho(\mathcal{F}) = \rho(A_{i_1} A_{i_2} \cdots A_{i_t})^{1/t}$$

for some finite product $A_{i_1}A_{i_2}\cdots A_{i_t} \in \mathcal{F}_t$, and the corresponding product sequence is called an optimal sequence. Within this framework, other special extremal norms, such as Barabanov norm [30], Optimal norm [22], etc., are also considered. Kozyakin in [19] considered an iterative algorithm which approximates $\rho(\mathcal{F})$ through constructing a sequence of approximated Barabanov norms by assuming irreducibility, however, the computational cost is very high and the issue of estimating the convergence rate remains unsolved. The sum of squares method investigated in [23] was intended to approximate the extremal norm by a multivariate polynomial with norm-like quality under which the action of matrices becomes contractive. However, to obtain an analytic extremal norm expression is quite challenging and there seem no satisfactory solutions so far.

The second approach makes use of the cone invariance of a given matrix set $\mathcal{F}$ for computing its joint spectral radius when such a property exists [24]. In [24, 25], an iterative algorithm building an approximated invariant set was developed, which for a fixed dimension demonstrates polynomial time complexity with respect to $1/\varepsilon$, where ε is a given accuracy. In [2], Blondel and Nesterov introduced a Kronecker lifting based approximation to the joint spectral radius with arbitrary accuracy under the assumption of the existence of an invariant proper cone, which can always be assured via one step of semi-definite lifting with the cost of squaring the matrix dimension. The exact nature of this cone is irrelevant to the derived accuracy of estimation. Following this methodology, a new conic programming method was offered in [27], which gives an improved accuracy estimation by taking the nature of cone invariance into the consideration. In general, the existence of an invariant cone is restrictive and may exclude many interesting cases in real applications.

The main contribution of our paper is to prove that any finite set of rank-one matrices satisfies the finiteness property by making use of Barabanov norm and rank-one property. As we know, rank-one matrices are the simplest class of matrices not only in theoretic analysis but also in algorithmic approximations for matrix computation since any matrix can be expressed in terms of the sum of a set of rank-one matrices, for example, by singular value decomposition. Among all those illustrative examples appeared in existing literature related to joint spectral radius, we observed that all rank-one cases satisfy the finiteness conjecture.

The paper is organized as follows. In section 2, we give some properties of rank-one matrices and prove that a finite set of rank-one matrices possesses the finiteness property by utilizing the well-known reduction lemma [1, 12] and Barabanov norm. In section 3, several numerical examples are presented to demonstrate our theoretical result. Concluding remarks are summarized in section 4.

2 Joint Spectral Radius of Rank-One Matrix Set

In this section, we will show that any finite set $\mathcal{F}$ of rank-one matrices possesses the finiteness property. Given a matrix $A \in \mathbb{C}^{n\times n}$, let rank$(A)$ be its rank. We know from linear algebra that rank$(A) = 1$ if and only if there exist two nonzero vectors $x, y \in \mathbb{C}^n$ such that $A = xy^*$. Obviously, any rank-one matrix A has at most one nonzero eigenvalue, denoted by $\lambda(A) = y^*x$. In particular, the spectral radius of a rank-one matrix A is $\rho(A) = |\lambda(A)|$. For any two rank-one matrices $A_1 = x_1y_1^* \in \mathbb{C}^{n\times n}$ and $A_2 = x_2y_2^* \in \mathbb{C}^{n\times n}$, the product

$$A_1A_2 = x_1y_1^*x_2y_2^* = (y_1^*x_2)x_1y_2^*$$

is at most rank-one. By a simple induction, arbitrary finite products of rank-one matrices remain at most rank-one.

If $\rho(\mathcal{F}) = 0$, then by (3) it holds $\rho(A_i) = 0 = \rho(\mathcal{F})$ for all $1 \le i \le m$ and so the finiteness property is already proved. Thus we will only consider the case with $\rho(\mathcal{F}) > 0$. Recall that a general matrix family $\mathcal{F}$ is said to be irreducible provided all the matrices in $\mathcal{F}$ have no common non-trivial invariant linear subspaces of $\mathbb{C}^n$. The following reduction lemma allows us to assume that $\mathcal{F}$ is irreducible in the following; for otherwise, we could first reduce $\mathcal{F}$ into several irreducible matrix families with smaller dimensions, and then carry out the same proof with each irreducible matrix family to draw the same conclusion.

Lemma 1 ([1]). *For any finite matrix family $\mathcal{F} = \{A_1, A_2, \cdots, A_m\} \subset \mathbb{C}^{n\times n}$, there exist a nonsingular matrix $P \in \mathbb{C}^{n\times n}$ and r positive integers $\{n_1, n_2, \cdots, n_r\}$ with $n_1 + n_2 + \cdots + n_r = n$ such that*

$$PA_iP^{-1} = \begin{bmatrix} A_i^{(1)} & 0 & \cdots & 0 \\ * & A_i^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & A_i^{(r)} \end{bmatrix} \quad \text{for} \quad i = 1, 2, \cdots, m,$$

where $\mathcal{F}^{(j)} := \{A_1^{(j)}, A_2^{(j)}, \cdots, A_m^{(j)}\} \subset \mathbb{C}^{n_j\times n_j}$ is irreducible for $j = 1, 2, \cdots, r$, satisfying

$$\rho(\mathcal{F}) = \max_{1\le j\le r} \rho(\mathcal{F}^{(j)}).$$

In particular, without lose of generality, we may always assume the matrix family $\mathcal{F}$ being irreducible. This leads to an important connection between the joint spectral radius and a special induced matrix norm, called extremal norm [18]. Due to the irreducibility of $\mathcal{F}$, we can assume $\rho(\mathcal{F}) = 1$ by normalizing the matrix family $\mathcal{F}$ by $1/\rho(\mathcal{F})$, and thus it guarantees that normalized $\mathcal{F}$ is non-defective, i.e., the semi-group of matrices generated by $\mathcal{F}$ is bounded, and hence there exists an (invariant) extremal norm for $\mathcal{F}$ as described in the next lemma.

Lemma 2 ([30]). *For any finite irreducible matrix family $\mathcal{F}$, there exists a vector (Barabanov) norm $\|\cdot\|_B$ such that:*

(1) For all $v \in \mathbb{C}^n$ and all $A \in \mathcal{F}$ it holds that $\|Av\|_B \le \rho(\mathcal{F})\|v\|_B$,
(2) For all $v \in \mathbb{C}^n$, there exists an $A \in \mathcal{F}$ such that $\|Av\|_B = \rho(\mathcal{F})\|v\|_B$.

In particular, the induced matrix norm $\|\cdot\|_B$ is an extremal norm satisfying

$$\max_{A\in\mathcal{F}} \|A\|_B = \rho(\mathcal{F}).$$

We are ready to prove the finiteness property for the irreducible case.

Theorem 3. *Let $\mathcal{F} = \{A_i = x_i y_i^* : i = 1, 2, \cdots, m\} \subset \mathbb{C}^{n\times n}$ be an irreducible rank-one matrix family. Then $\mathcal{F}$ has the finiteness property and the corresponding optimal product sequence of minimal length has distinct factors.*

Proof. We first normalize $\mathcal{F}$ such that $\rho(\mathcal{F}) = 1$. By Lemma 2, choose $v \in \mathbb{C}^n$ with $\|v\|_B = 1$, then for any $k \geq 1$ there exists a multi-index $(i_1, i_2, \cdots, i_k)$ such that

$$1 = \|v\|_B = \|A_{i_1} v\|_B = \|A_{i_2} A_{i_1} v\|_B = \cdots = \|A_{i_k} \cdots A_{i_2} A_{i_1} v\|_B. \qquad (4)$$

By the pigeonhole principle, if $k \geq (m+1)$, then the multi-index $(i_1, i_2, \cdots, i_k)$ has at least one repeated index. We define s to be the maximum of those k's such that the corresponding multi-index $(i_1, i_2, \cdots, i_k)$ satisfying (4) has no repetition. It's obvious that $s \leq m$. Then, choosing $k = s + 1$ in (4) gives $i_{s+1} = i_j$ for some unique $1 \leq j \leq s$, that is,

$$1 = \|v\|_B = \cdots = \|A_{i_j} \cdots A_{i_1} v\|_B = \cdots = \|A_{i_{s+1}} A_{i_s} \cdots A_{i_j} \cdots A_{i_1} v\|_B.$$

Since $A_{i_{s+1}} = A_{i_j}$ is rank-one matrix, its range is one-dimensional and hence

$$A_{i_j} \cdots A_{i_1} v = \alpha z \quad \text{and} \quad A_{i_{s+1}} A_{i_s} \cdots A_{i_j} \cdots A_{i_1} v = \beta z$$

for some $0 \neq \alpha \in \mathbb{C}, 0 \neq \beta \in \mathbb{C}$, and $0 \neq z \in \mathbb{C}^n$ (we may choose $z = x_{i_j}$ here). Then

$$\|\alpha z\|_B = \|A_{i_j} \cdots A_{i_1} v\|_B = 1 = \|A_{i_{s+1}} A_{i_s} \cdots A_{i_j} \cdots A_{i_1} v\|_B = \|\beta z\|_B,$$

which gives $|\alpha| = |\beta|$. Finally, we obtain

$$\beta z = A_{i_{s+1}} A_{i_s} \cdots A_{i_{j+1}} (A_{i_j} \cdots A_{i_1} v) = A_{i_{s+1}} A_{i_s} \cdots A_{i_{j+1}} (\alpha z)$$

and hence

$$A_{i_{s+1}} A_{i_s} \cdots A_{i_{j+1}} z = \frac{\beta}{\alpha} z,$$

where $\frac{\beta}{\alpha}$ is an eigenvalue of $A_{i_{s+1}} A_{i_s} \cdots A_{i_{j+1}}$. Therefore, by Lemma 2,

$$1 \geq \|A_{i_{s+1}} A_{i_s} \cdots A_{i_{j+1}}\|_B \geq \rho(A_{i_{s+1}} A_{i_s} \cdots A_{i_{j+1}}) \geq |\frac{\beta}{\alpha}| = 1,$$

which proves that $\mathcal{F}$ has the finiteness property with

$$\rho(\mathcal{F}) = 1 = \rho(A_{i_{s+1}} A_{i_s} \cdots A_{i_{j+1}})^{1/(s-j+1)},$$

where $1 \leq (s - j + 1) \leq m$ and $i_{s+1} \neq i_s \neq \cdots \neq i_{j+1}$ by the choice of s. □

We remark here that Theorem 3 provides us a critical structure of an optimal sequence, which will greatly improve the efficiency of specially designed search algorithms. In particular, non-repeated index indicates that the lengths of all minimal optimal sequences will not be longer than m, which is independent of the dimension of matrices. In fact, the possible minimal optimal sequence with longest length is $A_1 A_2 \cdots A_m$. In another way, an explicit formula for the joint spectral radius of any rank-one matrix family $\mathcal{F} = \{A_1, A_2, \cdots, A_m\} \subset \mathbb{C}^{n\times n}$ is

$$\rho(\mathcal{F}) = \max_{1\leq k\leq m} \left(\max_{A\in\mathcal{F}_k^{(*)}} \rho(A)^{1/k} \right), \tag{5}$$

where $\mathcal{F}_k^{(*)} = \{A_{i_1} A_{i_2} \cdots A_{i_k} \in \mathcal{F}_k : \ i_s \neq i_t \quad \text{when } s \neq t\}$ denotes all possible products in $\mathcal{F}_k$ with distinct factors.

3 Examples

In this section, we verify our formula (5) by two examples. The formula (5) provides a straightforward way to calculate the joint spectral radius for a rank-one matrix family. The search of all possible products with distinct factors of length not exceeding m is sufficient to obtain the exact value of $\rho(\mathcal{F})$, however, most of current numerical approximation methods can only provide lower and upper bounds for joint spectral radius with no indication whether the joint spectral radius has been achieved. Moreover, our formula (5) is fully validated by the reported optimal sequences for any pair of rank-one 2×2 sign-matrices in [6].

Example 1 ([6]). Consider the rank-one matrix pair

$$\mathcal{F} = \left\{ A_1 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \right\}.$$

Applying the formula (5) to obtain

$$\rho(\mathcal{F}) = \max_{1\leq k\leq 2} \max_{A\in\mathcal{F}_k^{(*)}} \rho(A)^{1/k} = \max\{\rho(A_1), \rho(A_2), \rho(A_1A_2)^{1/2}\} = \sqrt{2}.$$

While in [6] this was solved by constructing an extremal real polytope norm.

Example 2 ([14]). Consider the rank-one matrix family

$$\mathcal{F} = \left\{ A_1 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, A_3 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, A_4 = \begin{bmatrix} \frac{2}{3} & 0 \\ \frac{-2}{3} & 0 \end{bmatrix} \right\}.$$

Using the formula (5) to get

$$\rho(\mathcal{F}) = \max_{1\leq k\leq 4} \max_{A\in\mathcal{F}_k^{(*)}} \rho(A)^{1/k} = 1.$$

The same conclusion was derived in [14] by observing relations among all matrices, whose approach is difficult to be applied to general cases.

4 Concluding Remarks

In this paper, we show that any family of rank-one matrices possesses the finiteness property and an explicit formula of its joint/generalized spectral radius is obtained. Our next research target is to approximate the joint/generalized spectral of general matrix family by exploiting the rank-one approximation based on singular value decomposition.

Acknowledgment. The authors would like to thank Ahmadi and Parrilo for pointing out an error in section 2.1 of our earlier version posted on arXiv:1109.1356.

References

1. Berger, M.A., Wang, Y.: Bounded semigroups of matrices. Linear Algebra Appl. 166, 21–27 (1992)
2. Blondel, V.D., Nesterov, Y.: Computationally efficient approximations of the joint spectral radius. SIAM J. Matrix Anal. Appl. 27(1), 256–272 (2005)
3. Blondel, V.D., Nesterov, Y., Theys, J.: On the accuracy of the ellipsoid norm approximation of the joint spectral radius. Linear Algebra Appl. 394, 91–107 (2005)
4. Blondel, V.D., Tsitsiklis, J.N.: The boundedness of all products of a pair of matrices is undecidable. Systems Control Lett. 41(2), 135–140 (2000)
5. Bröker, M., Zhou, X.: Characterization of continuous, four-coefficient scaling functions via matrix spectral radius. SIAM J. Matrix Anal. Appl. 22(1), 242–257 (2000)
6. Cicone, A., Guglielmi, N., Serra-Capizzano, S., Zennaro, M.: Finiteness property of pairs of 2×2 sign-matrices via real extremal polytope norms. Linear Algebra Appl. 432(2-3), 796–816 (2010)
7. Dai, X., Huang, Y., Xiao, M.: Almost sure stability of discrete-time switched linear systems: A topological point of view. SIAM J. Control Optim. 47(4), 2137–2156 (2008)
8. Dai, X., Huang, Y., Xiao, M.: Criteria of stability for continuous-time switched systems by using liao-type exponents. SIAM J. Control Optim. 48(5), 3271–3296 (2010)
9. Daubechies, I., Lagarias, J.C.: Sets of matrices all infinite products of which converge. Linear Algebra Appl. 161, 227–263 (1992)
10. Daubechies, I., Lagarias, J.C.: Two-scale difference equations. II. Local regularity, infinite products of matrices and fractals. SIAM J. Math. Anal. 23(4), 1031–1079 (1992)
11. Daubechies, I., Lagarias, J.C.: Corrigendum/addendum to: “Sets of matrices all infinite products of which converge”. Linear Algebra Appl. 161, 227–263 (1992); Linear Algebra Appl. 327(1-3), 69–83 (2001)
12. Elsner, L.: The generalized spectral-radius theorem: An analytic-geometric proof. Linear Algebra Appl. 220, 151–159 (1995)
13. Gripenberg, G.: Computing the joint spectral radius. Linear Algebra Appl. 234, 43–60 (1996)
14. Guglielmi, N., Wirth, F., Zennaro, M.: Complex polytope extremality results for families of matrices. SIAM J. Matrix Anal. Appl. 27, 721–743 (2005)

15. Guglielmi, N., Zennaro, M.: Finding extremal complex polytope norms for families of real matrices. SIAM J. Matrix Anal. Appl. 31, 602–620 (2009)
16. Guglielmi, N., Zennaro, M.: An algorithm for finding extremal polytope norms of matrix families. Linear Algebra Appl. 428(10), 2265–2282 (2008)
17. Gurvits, L.: Stability of discrete linear inclusion. Linear Algebra Appl. 231, 47–85 (1995)
18. Jungers, R.M.: The joint spectral radius: theory and applications. Springer (2009)
19. Kozyakin, V.: Iterative building of barabanov norms and computation of the joint spectral radius for matrix sets. Discrete and Continuous Dynamical Systems - Series B 14(1), 143–158 (2010)
20. Lagarias, J.C., Wang, Y.: The finiteness conjecture for the generalized spectral radius of a set of matrices. Linear Algebra Appl. 214, 17–42 (1995)
21. Maesumi, M.: An efficient lower bound for the generalized spectral radius of a set of matrices. Linear Algebra Appl. 240, 1–7 (1996)
22. Maesumi, M.: Optimal norms and the computation of joint spectral radius of matrices. Linear Algebra Appl. 428(10), 2324–2338 (2008)
23. Parrilo, P.A., Jadbabaie, A.: Approximation of the joint spectral radius using sum of squares. Linear Algebra and its Applications 428(10), 2385–2402 (2008)
24. Protasov, V.Y.: The joint spectral radius and invariant sets of linear operators. Fundam. Prikl. Mat. 2(1), 205–231 (1996)
25. Protasov, V.Y.: The geometric approach for computing the joint spectral radius. In: Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference CDC-ECC 2005, pp. 3001–3006. IEEE Control Systems Society, Piscataway (2005)
26. Protasov, V.Y.: Fractal curves and wavelets. Izv. Ross. Akad. Nauk. Ser. Mat. 70(5), 123–162 (2006)
27. Protasov, V.Y., Jungers, R.M., Blondel, V.D.: Joint spectral characteristics of matrices: A conic programming approach. SIAM J. Matrix Anal. Appl. 31(4), 2146–2162 (2010)
28. Rota, G.C., Strang, G.: A note on the joint spectral radius. Nederl. Akad. Wetensch. Proc. Ser. A 63 = Indag. Math. 22, 379–381 (1960)
29. Shorten, R., Wirth, F., Mason, O., Wulff, K., King, C.: Stability criteria for switched and hybrid systems. SIAM Rev. 49(4), 545–592 (2007)
30. Wirth, F.: The generalized spectral radius and extremal norms. Linear Algebra Appl. 342(1-3), 17–40 (2002)

Novelty Detection Using a New Group Outlier Factor

Amine Chaibi, Mustapha Lebbah, and Hanane Azzag

University of Paris 13, Sorbonne Paris City - CNRS
LIPN-UMR 7030
99, av. J-B Clement - F-93430 Villetaneuse
{firstname.secondname}@lipn.univ-paris13.fr

Abstract. We present in this paper a new measure named GOF (Group Outlier Factor) for cluster outliers and novelty detection. The main difference between GOF and existing methods is that being an outlier is not associated to a single pattern but to a cluster. GOF is based on relative density of each group of data and provides a quantitative indicator of outlier-ness which enables to detect automatically ”cluster outliers”. To learn GOF measure, we integrate it in a clustering process using Self-organizing Map. Experimental results and comparison studies show that the use of GOF sensibly improves the results in term of cluster-outlier detection and novelty detection.

Keywords: novelty detection, group outliers, outliers, clustering, SOM.

1 Introduction

Outlier detection has attracted increasing attention in machine learning and data mining field due to the numerous applications, including credit card fraud detection, network intrusion and the discovery of criminal activities in electronic commerce. Outlier detection is a data mining task whose purpose is to isolate the patterns which are dissimilar from the remaining data. There is a strong synergy between outlier detection and novelty detection that becomes important in machine learning. It has been confirmed in several studies that novelty detection is an extremely challenging task [1]. Recently, many studies have been conducted on outlier detection for large datasets [3,2]. Most of them consider being an outlier pattern as a binary property. For many applications, the situation is more complex and it becomes useful to assign to a cluster a degree of being an outlier.

In this paper, we introduce a new method for cluster outliers detection in multidimensional dataset that assigns a Group Outlier Factor (GOF) score to each cluster in the aim to measure a degree of outlier-ness. This is, to the best of our knowledge, the first concept which quantifies how outlying a cluster is. The main difference between our approach and existing methods is that being an outlier is not associated to a single pattern but to a cluster. Our approach must not be confused with Local Outlier Factor (LOF) method [4][5]. LOF indicates the degree of outlier-ness for each pattern by comparing the local density of an observation with the average density of its k-nearest neighbors (*k-NN*) “without learning”.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 364–372, 2012.

To learn the particularity of each group in the data distribution, we incorporate GOF parameter into a training process of clustering algorithm. For this purpose, we use Self-Organizing Maps (SOM) algorithm. SOM is a useful tool to visualize and cluster high-dimensional data in low-dimensional views [6]. Our method is an hybrid approach, which combines clustering and density based approaches.

In this work, we are also interested to novelty detection problem. Understanding when new data are novel can be extremely important in order to automatically detect outlier clusters and novel patterns. In this paper, we detail the use of GOF measure for novelty detection. Thus, we propose two algorithms named GOF-SOM, and GOF-Novelty that offer respectively a detection of outlier cluster and novelty detection.

The remaining of this paper is organized as follows : we briefly review the related works in section 2. In section 3, we present the model and algorithms. Section 4 is devoted to the methodology and experimental results. Finally, section 5 concludes this work and proposes some perspectives.

2 Related Works

In this section, we discuss previous works on outlier and novelty detection problems. Different methods for novelty detection are reviewed in [1,7]. Among them, we cite statistical based approaches that are mostly based on modeling data by there statistical properties, as density to estimate whether a samples comes from the same distribution or not [1]. In statistical based approaches, two methods to estimate the probability density function exist : the parametric and non-parametric approaches. Parametric approaches make an assumption that data distributions are Gaussian [8]. However, non-parametric approaches do not make any assumption on the statistical properties of data. To estimate the density of multidimensional data, one way is to use nearest neighbour based on density estimation or parzen density estimation [9].

In [10], author introduces a kernel PCA for novelty detection. The main idea of this approach is to assign training data into a high dimensional feature space where kernel PCA extracts the principal components of the data distribution. The squared distance to the corresponding principal subspace is used to measure novelty. Support Vector Machines (SVM) are also used for novelty detection [11,12]. Indeed, a support vector algorithm was used to characterize the support of a high dimensional distribution. With this algorithm, one can compute a set of contours which encloses the data. These contours can be considered as normal data boundaries. The data outside the boundaries are interpreted as novelties. Other works propose self-organizing map approach for spatial outlier detection [13,14]. In [15], authors transfer the unsupervised learning of outlier detection to the non-parameter regression learning and propose a multi-scale local kernel regression method by combining informations of the multiple scale neighborhoods to compute the outlier factors.

There are several machine learning algorithms in the literature that define outlier as observation. However, they do not consider that a cluster can being outlier. Therefore, in this work, we provide a new formal definition of group outliers, which avoids the shortcomings present in traditional approaches. A group outliers is a set of patterns forming a cluster considerably isolated from the rest of clusters.

The motivation of this paper is to describe a new concept of ”cluster outlier-ness”. In order to quantify it, we propose a relative measure of isolation called Group Outlier Factor (GOF). GOF is a score, wich is computed during a clustering process using self-organizing maps. Thus, an outlier factor with respect to each cluster is computed for each new sample and compared to the GOF parameter associated for each cluster. If the outlier factor of pattern is much greater than GOF of the corresponding cluster, the sample is classified as novel. This approach allows to identify meaningful outlier-clusters and detects a novel data that previous approaches could not find.

3 Proposed Algorithm

In this section, we present a new approach for cluster-outlier detection and its application for novelty detection. Given a set of training data $\mathcal{D} = \{\mathbf{x}_i; i = 1, \ldots, N\}$, where each observation $\mathbf{x}_i = (x_i^1, x_i^2, ..., x_i^d)$ is a vector in $\Re^d$. The main idea of our cluster-outlier technique is to simultaneously cluster data and learn a new parameter of cluster oulier-ness. In many applications, it is reasonable to assume that some patterns are not outliers alone but placed together with others can be considered as outlier-cluster.

3.1 GOF-SOM: Group Outlier Factor and Self-Organizing Map

In GOF-SOM, we use self-organizing maps (SOM) as clustering algorithm [6], which is increasingly used as tools for clustering and visualization. SOM consists of a discrete set of cells called map with size $\mathcal{C}$. This map has a discrete topology defined as an undirected graph, it is usually a regular grid in 2 dimensions. For each pair of cells (c,r) on the map, the distance $\delta(c, r)$ is defined as the length of the shortest chain linking cells r and c on the grid. For each cell c this distance defines a neighbor cell.

In GOF-SOM, each cell c of the grid $\mathcal{C}$ is associated with two parameters : a prototype vector $\mathbf{w}_c = (w_c^1, w_c^2..., w_c^j, ..., w_c^d)$ and a new parameter $GOF_c \in \Re$ (Group Outlier Factor). For each pattern $\mathbf{x}_i$, a local density $f_c(\mathbf{x}_i)$ is computed, which is defined as follows :

$$f_c(\mathbf{x}_i) = \exp^{-\frac{\|\mathbf{w}_c - \mathbf{x}_i\|^2}{2\sigma^2}}$$

where σ is the deviation of data.

After computing a density $f_c(\mathbf{x}_i)$ of pattern $\mathbf{x}_i$, we estimate a Group Outlier Factor (GOF), for each cluster denoted by P_c associated to a cell c. The larger is the value of GOF, the more probably the cluster is outlier. Thus, using topological maps, we propose to minimize the new following cost function :

$$\mathcal{R}(\mathcal{W}, \phi, GOF) = \sum_{i=1}^{N} \sum_{c=1}^{\mathcal{C}} K(\delta(\phi(\mathbf{x}_i), c)) \|\mathbf{w}_c - \mathbf{x}_i\|^2 + \sum_{i=1}^{N} \sum_{c=1}^{\mathcal{C}} K(\delta(\phi(\mathbf{x}_i), c)) \left(GOF_c - \frac{\frac{\sum_{\mathbf{x}_j \in P_c} \frac{1}{f_c(\mathbf{x}_j)}}{|P_c|}}{\frac{1}{f_c(\mathbf{x}_i)}} \right)^2 \quad (1)$$

where ϕ assigns each observation $\mathbf{x}_i$ to a single cell of the map. We denote by $\mathcal{W} = \{\mathbf{w}_c, \mathbf{w}_c \in \Re^d\}_{c=1}^{C}$ the set of prototypes and $GOF = \{GOF_1, ..., GOF_C\}$ the set of outlier-ness indicators.

The first term of the cost function (1) depends on the parameters $\mathcal{W}$ and ϕ, which enables to estimate the prototypes. The second term depends on the cluster outlier factor (GOF) associated to each cell. The minimization of $\mathcal{R}(\mathcal{W}, \phi, GOF)$ is run by iteratively performing three steps presented in algorithm 1.

3.2 GOF-Novelty: Group Outlier Factor for Novelty Detection

Based on the new definition of group outlier, the method that we propose for novelty detection works in two steps as follows :

Algorithm 1. GOF-SOM Algorithm

```
1: Inputs : The data $\mathcal{D} = \{\mathbf{x_i}\}_{i=1..N}$. A map with $C$ cells. Initialized prototypes $\mathcal{W} = \{\mathbf{w_c}, c = 1..C\}$, and GOF values for each cell $c$. $t_{max}$ : the maximum number of iterations.
2: Outputs : A partition $P = \{P_c\}_{c=1..C}$. The GOF values $= \{GOF_c, c = 1..C\}$.
3: Competition phase : Assign data $\mathbf{x}_i$ by using the function
       $$\phi(\mathbf{x}_i) = \arg \min_{1 \leq c \leq C} \| \mathbf{x}_i - \mathbf{w}_c \|^2$$
4: Adaptation phase (for each cell $c$) :
     – Update prototypes $\mathbf{w}_c$ : $\mathbf{w}_c(t) = \mathbf{w}_c(t-1) - \varepsilon(t)K(\delta(\phi(\mathbf{x}_i), c)))\,(\mathbf{w}_c(t-1) - \mathbf{x}_i)$
     – Update the values of $GOF_c$ :
       $$GOF_c(t) = GOF_c(t-1) - \varepsilon(t)K(\delta(\phi(\mathbf{x}_i, c))) \left( GOF_c(t-1) - \frac{\frac{\sum_{\mathbf{x}_j \in P_c} \frac{1}{f_c(\mathbf{x}_j)}}{|P_c|}}{\frac{1}{f_c(\mathbf{x}_i)}} \right)$$
       where $\varepsilon(t)$ is the step of learning.
5: Repeat phases 3 and 4 up to $t = t_{max}$.
```

- *GOF-SOM learning :* In the first step, we train GOF-SOM map, which provides a new parameter by computing an outlier-ness value for each cluster c denoted by GOF_c. If patterns assigned to the same cluster are dense and isolated, it would have a high degree of being outlier group. The advantage of GOF-SOM algorithm is to provide a clustering and a topological structure for outlier visualization task.
- *Novelty detection :* In the second step, we build a classifier for novelty detection using the GOF parameter provided by GOF-SOM. We define algorithm 2 called GOF-Novelty where the main function consists on assigning the new patterns $\mathbf{x}_i$ using GOF parameter. Thus, for each observation, we compute an Outlier Factor ($OF_c(\mathbf{x}_i)$) associated to cluster c as follows :

Algorithm 2. GOF-Novelty Algorithm

```
1: Inputs : A partition $P = \{P_c\}_{c=1..C}$. GOF values's = $\{GOF_c, c = 1..C\}$. The new dataset $\mathcal{D}^{'} = \{x_i\}_{i=1..M}$.
2: Outputs : $Novelty\_label$ : vector contains a binary value which indicates the novelty.
3: for i=1 : $M$ do
4:     $OF_{\phi(\mathbf{x}_i)}(\mathbf{x}_i) = \frac{\frac{\sum_{\mathbf{x}_j \in P_{\phi(\mathbf{x}_i)}} \frac{1}{f_c(\mathbf{x}_j)}}{|P_{\phi(\mathbf{x}_i)}|}}{\frac{1}{f_c(\mathbf{x}_i)}}$
5:     $Dif = abs\left(OF_{\phi(\mathbf{x}_i)}(\mathbf{x}_i) - GOF_{\phi(\mathbf{x}_i)}\right)$
6:     if | $Dif < threshold$ | then
7:         $Novelty\_label(\mathbf{x}_i) = 0;$
8:     else
9:         $Novelty\_label(\mathbf{x}_i) = 1;$
10:    end if
11: end for
   # The threshold is defined by the deviation of the output GOF, but it can vary for each dataset.
```

$$OF_c(\mathbf{x}_i) = \frac{\frac{\sum_{\mathbf{x}_j \in P_c} \frac{1}{f_c(\mathbf{x}_j)}}{|P_c|}}{\frac{1}{f_c(\mathbf{x}_i)}}$$

If the value of $OF_c(\mathbf{x}_i)$ is largely higher than the Outlier Factor of cluster (GOF_c), the data are necessarily novel.

4 Experimentation

This section details experiments carried out to assess the performance of GOF measure and GOF-Novelty. Synthetical, public and One-class classification datasets are used

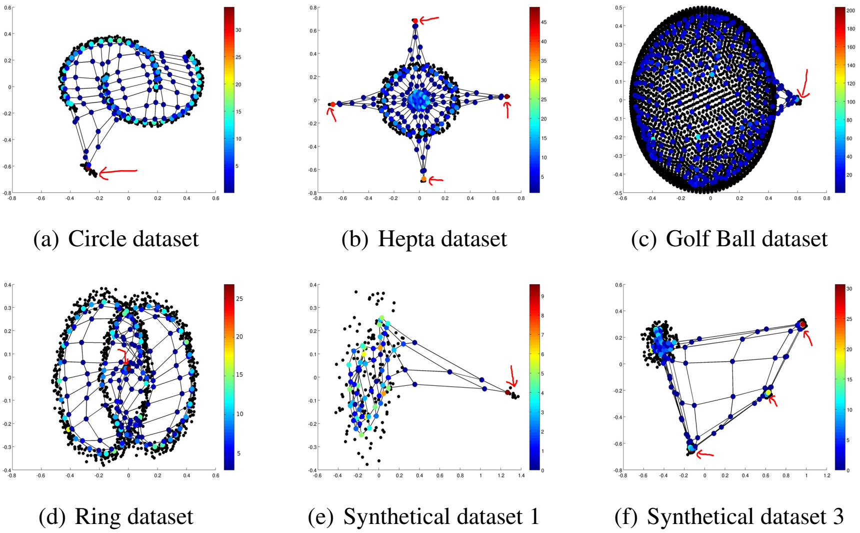
Table 1. Characteristics of the synthetical, public and One-class classification datasets

Public and synthetical datasets					One-class classifier datasets				
Datasets	Size	# Features	# No outlier	# Outliers	Datasets	Size	# Features	# No outlier	# Outliers
Ring	1072	2	943	129	Iris Setosa	150	4	50	100
Circle	638	3	586	52	Sonar Mines	108	60	11	97
Hepta	212	2	136	76	Biomed Healthy	194	5	127	67
Lsun	400	2	300	100	Hepatitis Normal	155	19	123	32
Target	951	2	787	164	Diabetes Present	768	8	500	268
Golf Ball	4343	3	3941	402	Ecoli Periplasm	336	7	52	284
Synthetical dataset 1	160	4	143	17	Spectf 1	349	44	254	95
Synthetical dataset 2	234	4	208	26	Balance-Scale	625	4	288	337
Synthetical dataset 3	569	4	357	212	Glass Building	214	9	70	144
Synthetical dataset 4	402	4	292	110	Waveform 2	900	21	300	600

in the experimentaions (Table 1) [16]. For synthetical datasets, we generate a small clusters isolated from the rest of the data. All the datasets used for this task are in 2-dimensional and presented in table 1. One-class classification (occ) datasets [17] are extracted from : http://homepage.tudelft.nl/n9d04/occ/index.html.

4.1 Analysis of GOF Measure

To learn the behavior of GOF measure, we visualize the mapping and the GOF parameter learned by GOF-SOM. This is a key factor in supporting the analysts in the analysis task. Thus, they can spot cluster-outliers, which might be errors in the manual labeling, or a characteristic of dataset structure. Figure 1, displays the dataset and the learned map. The GOF score is indicated with color degree, which reflects the outlier-ness of each cluster (cell of the map). The cluster outliers can be identified by dark red color (the more the prototype's color is red, the more the cluster is outlier). It is clear for the presented dataset that red color corresponds to the isolated cluster with high value of GOF (the group outlier is indicated with an arrow).



(a) Circle dataset

(b) Hepta dataset

(c) Golf Ball dataset

(d) Ring dataset

(e) Synthetical dataset 1

(f) Synthetical dataset 3

Fig. 1. GOF-SOM map and dataset

4.2 GOF-Novelty: Comparative Study

The datasets used in this experimentaion have two classes : no outlier labled 0 and outlier labled 1. In the case of synthetical and public datasets, the outlier class is represented by the minority class in the datasets. For occ datasets, the classification (outlier, no outlier) is already defined by the author of datasets [17]. In this experimentation, a training dataset contains only data labeled 0 (no outlier). Test dataset contains data labeled 1 (outlier) and 20% of data have label 0. The criteria Recall, Precision and AUC are used for evaluation [18]. The Recall or True Positive Rate is the proportion of positive cases correctly identified. Precision is the proportion of the predicted positive cases that are correct. AUC is the Area Under the ROC curve. It is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. We compare SOM-Novelty with One-SVM and PCA approaches. The experimental results are shown in table 2.

Table 2. Results obtained with GOF-Novelty PCA, and One-SVM on Recall, Precision and AUC

Datasets	Recall index			Precision index			AUC index		
	GOF-Novelty	PCA	One-SVM	GOF-Novelty	PCA	One-SVM	GOF-Novelty	PCA	One-SVM
Synthetical dataset 1	1	0.35	0.85	1	1	0.86	1	0.68	0.43
Synthetical dataset 2	1	0.33	0.84	1	1	1	0.95	0.66	0.92
Synthetical dataset 3	0.53	0.5	0.52	1	1	1	0.76	0.75	0.76
Synthetical dataset 4	0.59	0.24	0.52	1	1	1	0.8	0.62	0.76
Circle	0.53	0.21	0.27	1	1	1	0.83	0.6	0.63
Ring	0.83	0.8	0.07	1	0.91	0.31	0.91	0.4	0.03
Lsun	0.86	0.66	0.3	1	0.72	1	0.68	0.44	0.65
Hepta	0.76	0.55	0.24	1	1.00	1	0.88	0.78	0.62
GolfBall	0.89	0.67	0.01	1	0.89	1	0.94	0.34	0.5
Iris Setosa	1	0.52	0.92	0.81	1	1	0.95	0.76	0.96
Sonar Mines	0.71	0.67	0.35	0.56	0.56	0.35	0.54	0.53	0.49
Biomed Healthy	0.94	0.35	0.85	0.97	0.94	0.61	0.67	0.65	0.66
Hepatitis Normal	0.69	0.46	0.87	0.8	0.86	0.66	0.52	0.59	0.51
Diabetes Present	0.99	0.53	0.71	0.73	0.65	0.65	0.52	0.5	0.5
Ecoli Periplasm	0.85	0.33	0.88	0.4	0.14	0.42	0.8	0.48	0.89
Spectf 1	0.85	0.45	0.73	0.71	0.69	0.72	0.52	0.46	0.49
Balance-Scale Left	0.86	0.44	0.56	0.44	0.45	0.43	0.51	0.49	0.47
Glass Building Float	0.9	0.13	0.66	0.31	0.29	0.27	0.75	0.49	0.61
Waveform 2	0.63	0.53	0.45	0.45	0.31	0.35	0.53	0.47	0.62

Table 2 present experimental results of Recall, Precision and AUC indexes. The main criteria for evaluating novelty detection is the maximization of detecting true novel patterns (Recall) [19]. GOF-Novelty provides the highest Recall values for all datasets, except for Hepatitis Normal and Ecoli Periplasm datasets. We observe a slight decrease of Recall if we compare our approach with the best one (PCA or One-SVM). Concerning Precision index, GOF-Novelty provides the best values for the most datasets. For Iris Setosa and Hepatitis Normal datasets GOF-Novelty is slightly less efficient. Despite a low decrease of both criteria in some datasets, GOF-Novelty provides a stable results for most datasets. Thus, in Ring database, One-SVM provides precision 0.31 where

GOF-Novelty provides 1. For Ecoli Periplasm, GOF-Novelty provides 0.40 and PCA provides the lowest value 0.14. Observing AUC index, GOF-Novelty gives the highest AUC values in the majority except for Ecoli Periplasm and Waveform 2 where our approach is better than PCA. Despite a low decrease of AUC index in some datasets, GOF-Novelty provides a stable results. For example in Circle, GOF-Novelty obtains 0.91, but PCA obtains 0.4, One-SVM decreases and provides 0.03. Comparing to other methods, we clearly see that GOF-Novelty is a good way to detect a novelty and provides a measure of outlier-ness for each cluster.

5 Conclusion and Perspectives

This paper studies the problem of cluster-outlier detection and its application to novelty detection. To our knowledge, this paper is the first work that gives a new definition of Group Outlier Factor (GOF), which measures the degree of being cluster outlier. We have used GOF parameter more tightly with clustering in order to use it for novelty detection. A series of experiments were conducted to validate the proposed method which uses GOF parameter on Self-Organizing Map and compute many criteria to show the behavior of GOF in novelty detection. These results demonstrate that our method is promising and identify meaningful outlier-clusters and detect a novel data that previous approaches could not find. There are many perspectives to study after this results. The first one consists on further improving the performance of the GOF computation. Secondly, we will improve the usefulness of GOF by integrating it to time series and conducting more extensive tests with external indexes.

Acknowledgment. This research was supported by Anticipeo company.

References

1. Markou, M., Singh, S.: Novelty detection: a review part 1: statistical approaches. Signal Process. 83, 2481–2497 (2003)
2. Angiulli, F., Ben-Eliyahu-Zohary, R., Palopoli, L.: Tractable strong outlier identification. CoRR **abs/1109.4623** (2011)
3. Su, X., Tsai, C.L.: Outlier detection. Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery 1(3), 261–268 (2011)
4. Breunig, M., Kriege, H., Ng, R., Sander, J.: Lof: Identifying density-based local outliers. In: ACM SIGMOD 2000 International Congerence on Management of Data (2000)
5. Hasan, M.A., Chaoji, V., Salem, S., Zaki, M.J.: Robust partitional clustering by outlier and density insensitive seeding. Pattern Recogn. Lett. 30, 994–1002 (2009)
6. Kohonen, T., Schroeder, M.R., Huang, T.S. (eds.): Self-Organizing Maps, 3rd edn. Springer-Verlag New York, Inc., Secaucus (2001)
7. Markou, M., Singh, S.: Novelty detection: a review part 2: neural network based approaches. Signal Process. 83, 2499–2521 (2003)
8. Hansen, L.K., Liisberg, C., Salamon, P.: The error-reject tradeoff. Open Systems and Information Dynamics 4, 159–184 (1995)
9. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley (2001)
10. Hoffmann, H.: Kernel pca for novelty detection. Pattern Recognition 40 (2007)

11. Vapnik, V.N.: The nature of statistical learning theory. Springer, New York (1995)
12. Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. 13(7), 1443–1471 (2001)
13. Cai, Q., He, H., Man, H.: Somso: A self-organizing map appoach for spacial outlier detection with multiple attributes. In: Proccedings of IJCNN 2009, pp. 425–431 (2009)
14. Cai, Q., He, H., Man, H., Qiu, J.: Iterativesomso: An iterative self-organizing map for spatial outlier detection. In: Proccedings of IJCNN 2010, pp. 325–330 (2010)
15. Gao, J., Hu, W., Li, W., Zhang, Z.: Local outlier detection based on kernel regression. In: International Conference on Pattern Recognition (2010)
16. Frank, A., Asuncion, A.: Uci machine learning repository. Technical report, University of California, Irvine, School of Information and Computer Sciences (2010)
17. Moya, M.M., Hush, D.R.: Network constraints and multi-objective optimization for one-class classification. Neural Netw. 9, 463–474 (1996)
18. Fawcett, T.: An introduction to roc analysis. Pattern Recognition Letters 27(8), 861–874 (2006)
19. Yeung, D.Y., Chow, C.: Parzen-window network intrusion detectors. In: Proceedings of the Sixteenth International Conference on Pattern Recognition, pp. 385–388 (2002)

Hierarchical K-Means Algorithm for Modeling Visual Area V2 Neurons

Xiaolin Hu, Peng Qi, and Bo Zhang

State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), and Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
xiaolin.hu@gmail.com, pengrobertqi@163.com, dcszb@tsinghua.edu.cn

Abstract. Computational studies about the properties of the receptive fields of neurons in the cortical visual pathway of mammals are abundant in the literature but most addressed neurons in the primary visual area (V1). Recently, the sparse deep belief network (DBN) was proposed to model the response properties of neurons in the V2 area. By investigating the factors that contribute to the success of the model, we find that a simple algorithm for data clustering, K-means algorithm can be stacked into a hierarchy to reproduce these properties of V2 neurons, too. In addition, it is computationally much more efficient than the sparse DBN.

Keywords: Neural network, Deep learning, Visual area, V1, V2.

1 Introduction

Since Hubel and Wiesel [1] found that the receptive fields of many neurons in the primary visual cortex (V1) are edge detectors, a wealth of researches have attempted to interpret this ground breaking discovery. Two well-known proposals refer to sparse coding [2,3] and independent component analysis (ICA) [4]. Both approaches can be understood as a single layer network where the inputs are image pixels and the outputs correspond to the responses of V1 simple cells, which are assumed to be sparse, i.e., the output units should keep silence or near silence most of the time and fire only occasionally. Sparsity is closely related to high-order statistics of natural images, which plays a significant role in reproducing the edge-like structure of the receptive fields of V1 simple cells. In fact, with sparsity constraint many other models such as the restricted Boltzmann machine (RBM) [5], auto-encoder [6] and K-means algorithm [7,8] have been found to be able to learn the edge-like structure of the receptive fields of V1 neurons on natural images.

Hierarchical models [9,10] have been proposed for modeling the response properties of V1 complex cells, another important type of neurons in V1 area. However, there have been few attempts to quantitatively model the properties of neurons beyond V1 along the cortical visual pathway such as V2 or V4. The famous hierarchical model HMAX [11] was tested against V4 neurons and achieved remarkable results [12]. But the properties of its low level units are handcrafted

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 373–381, 2012.

and what is more interesting to the computational neuroscience community is learning each layer in a similar way. The deep belief network [13] is such a model. It consists of multiple layers of RBMs, and learning starts from the bottom layer to the top layer in the sequel. It was found that a two-layer DBN is able to replicate some properties of the receptive fields of both V1 and V2 neurons by imposing a sparse firing constraint on each layer [5]. This model owes its success largely to its nonlinearity on the the first layer output. In the present paper, we will show that the difference between the sparsity degrees on the two layers are also critical for producing these results. To be more specifically, the second layer firing should not be as sparse as the first layer. If one seeks alternative models for doing similar task, neither of the two factors should be ignored.

In the paper, we will show that the K-means algorithm, a simple data clustering algorithm, can be stacked into a hierarchy to model V2 neurons. However, as the standard K-means algorithm is an extremely sparse model (for each input data only one hidden unit fires), to control its sparsity degree, some modifications are needed.

2 Sparse Deep Belief Network

A restricted Boltzmann machine (RBM) consists of a layer of visible units $\mathbf{v}$, a layer of hidden units $\mathbf{h}$ and a symmetric connections weights between the two layers represented by a matrix W. The visible units and hidden units have biases, denoted by c_i and b_j, respectively [14]. The sparse RBM imposes a sparse firing constraint on the hidden units [5]. With a set of training data $\mathbf{v}_1, \ldots, \mathbf{v}_N$ where $\mathbf{v}_n \in R^D$, the sparse RBM minimizes the following function

$$-N\langle \log \sum_{\mathbf{h}} P(\mathbf{v}, \mathbf{h})\rangle + \lambda \sum_{j=1}^{K} \|p - \langle E(h_j|\mathbf{v})\rangle\|^2$$

over w_{ij}, c_i and b_j, where

$$-\log P(\mathbf{v}, \mathbf{h}) = \frac{1}{2\sigma^2}\sum_i v_i^2 - \frac{1}{\sigma^2}\left(\sum_i c_i v_i + \sum_j b_j h_j + \sum_{i,j} v_i w_{ij} h_j\right)$$

and $\lambda, \sigma > 0$. In above equations, $\langle\cdot\rangle$ denotes average over samples and $E(\cdot)$ denotes the conditional expectation given the data. The parameter p is the desired firing probability of the hidden units, which controls the sparsity degree of firing.

With a modified contrastive divergence learning rule [5], the sparse RBM can learn the gabor-like receptive fields of V1 simple cells on natural images. Fig. 1 visualizes 200 weights associated with the hidden units. They were learned on a large set of randomly selected 14-by-14 patches from ten 512-by-512 natural images [2], which were preprocessed by $1/f$ whitening and low pass filtering in the frequency domain. The sparsity parameter is set as $p = 0.02$.

We stacked another sparse RBM with 200 hidden units on top of the first layer, and trained the second layer weights and biases by freezing the first layer

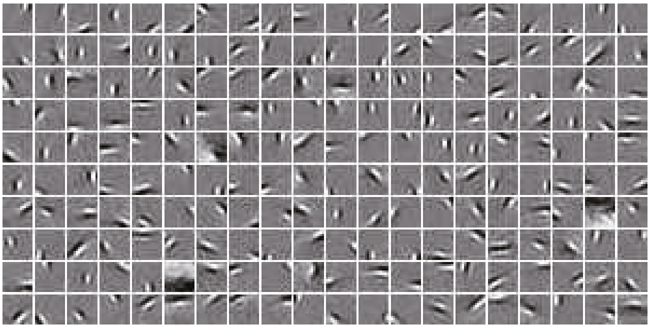
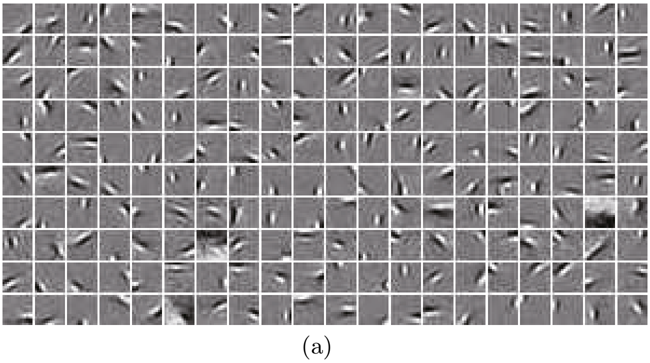
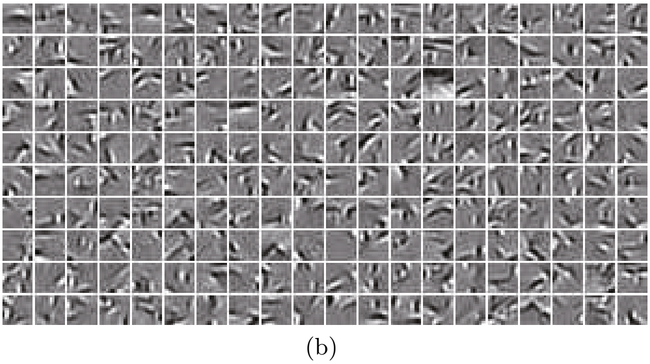


Fig. 1. Visualization of 200 first layer weight vectors of the sparse DBN. Each 14 × 14 patch corresponds to a weight vector.



(a)



(b)

Fig. 2. Visualization of 200 second layer weight vectors of the sparse DBN. (a) $p = 0.02$, (b) $p = 0.04$.

weights and biases. The resulting model is called *sparse deep belief network* or sparse DBN [5]. The receptive fields of the second layer units are visualized in Fig. 2 as weighted sum of the receptive fields of first layer units. It is seen that with $p = 0.02$ the receptive fields are visually similar to the receptive fields of the first layer; while with $p = 0.04$ the receptive fields are like edge conjunctions or corners, in agreement with the V2 neuron properties. In fact, with increasing p (greater than 0.02), our experiments showed that the structure of the receptive fields became more and more complex (data not shown). This observation suggests that the nonlinearity of the sparse RBM is not the only factor that contributes to the emergence of V2 neuron receptive fields, and the higher firing rate on the second layer than on the first layer is another critical factor. If one seeks alternative models for reproducing the V2 neuron properties, both factors should be considered.

3 Hierarchical K-Means Algorithms

3.1 K-Means Algorithms

The goal of K-means algorithm is to partition the data set $\mathbf{v}_1, \ldots, \mathbf{v}_N$ into K clusters. If we introduce a latent variable $\mathbf{w}_j$, the mean or centroid of cluster j, where $j = 1, \ldots, K$, then the goal is to identify $\mathbf{w}_j$. The algorithm consists of two iterative steps:

- For each input $\mathbf{v}_n$ determine which cluster it belongs to. Mathematically, this amounts to determine $j^* = \arg\min_j \|\mathbf{v}_n - \mathbf{w}_j\|$.
- Update $\mathbf{w}_j$ for $j = 1, \ldots, K$ by taking the mean (centroid) of data assigned to cluster j.

Each data point $\mathbf{v}_n$ is assigned a binary indicator vector $\mathbf{h}$ where $h_j = 1$ if this point belongs to cluster j and $h_j = 0$ otherwise. If the latent variables h_j are viewed as “neurons”, then the firing pattern of these neurons is extremely sparse—for each input only one neuron fires.

3.2 Multiple Firing K-Means Algorithms

Now we relax the algorithm by allowing multiple hidden units fire together for an input in the first step. Specifically, for each input $\mathbf{v}_n$ we determine L clusters it belongs to. Mathematically, this amounts to determine a set $\Omega \subset V = \{1, \ldots, K\}$ such that $|\Omega| = L$ and $\|\mathbf{v}_n - \mathbf{w}_s\| \leq \|\mathbf{v}_n - \mathbf{w}_j\|$ for $s \in \Omega$ and $j \in V \backslash \Omega$.

For each input $\mathbf{v}_n$ set

$$h_j(\mathbf{v}_n) = \begin{cases} 1, & \text{if } \mathbf{v}_n \text{ belongs to cluster } j; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Then there are always L hidden units firing. For this reason this algorithm is called *multiple firing K-means algorithm*. Its convergence results are stated in the following theorem.

Theorem 1. *Each step of the multiple firing K-means algorithm lowers the value of the function*

$$J = \langle \sum_{j=1}^{K} h_j \|\mathbf{v} - \mathbf{w}_j\|^2 \rangle \tag{2}$$

until convergence.

Proof. In step 1, $\mathbf{w}_j$ is fixed. It is easy to see that setting $h_j = 1$ for $j \in \Omega$ and $h_j = 0$ for $j \in V \backslash \Omega$ corresponds to the minimum of J over the binary vector $\mathbf{h}$ subject to the constraint that for each input there are always L elements equal to 1. In step 2, $\mathbf{h}$ is fixed. Notice that $\frac{\partial J}{\partial \mathbf{w}_j} = -2\langle h_j(\mathbf{v} - \mathbf{w}_j)\rangle$. Then step 2 is equivalent to taking $\frac{\partial J}{\partial \mathbf{w}_j} = 0$, which corresponds to minimization of J over $\mathbf{w}_j$. Therefore, each step results in a decrease of J until convergence.

3.3 Hierarchical Model

Similar to the sparse DBN, we can stack another multiple firing K-means algorithm on top of the first layer. It takes the output of the first layer as input and learns the centroids of the inputs by freezing the first layer centroids.

4 Experiments

It has been shown that the standard K-means algorithm can reproduce the gabor-like receptive fields of V1 cells [7,8]. Here we show that the multiple firing K-means algorithm has the same capability. A large number of 14-by-14 patches were randomly extracted from ten natural images, which were preprocessed in the same way as in Section 2. At every iteration 50,000 patches were input to the network and the centroids got updated once. After about 100 iterations the algorithm converged. The 200 centroids are plotted in Fig. 3 with $L = 3$. For other small values of L the results were visually similar to this figure (we tested $L = 5, 7, 10$).

We stacked a second layer multiple firing K-means algorithm to the output of the first layer. The second layer had 200 units and L was set to 10. After 100 iterations, the algorithm converged. It was found that only a few elements in the learned second layer centroids were significantly larger than zero (data not shown). The second layer centroids are visualized in Fig. 4 in the same manner as Fig. 2. It is seen that the shape of many second layer centroids are like corners or conjunctions of edges, in agreement with some V2 neurons properties [5].

To test the properties of the second layer units, we generated a set of angle stimuli as shown in Fig. 5 [15]. Each stimulus was a 14-by-14 image patch representing an angle in $\{\frac{2\pi}{M}, \frac{4\pi}{M}, \ldots, \frac{2(M-1)\pi}{M}\}$ in different orientations, which resulted in $M(M-1)$ different stimuli. See [15] for details. In addition, each stimulus was normalized to zero mean and unit variance. To identify the "center" of each second layer unit's receptive field, we translated all stimuli densely over the 14×14 input image patch, and identified the position at which the

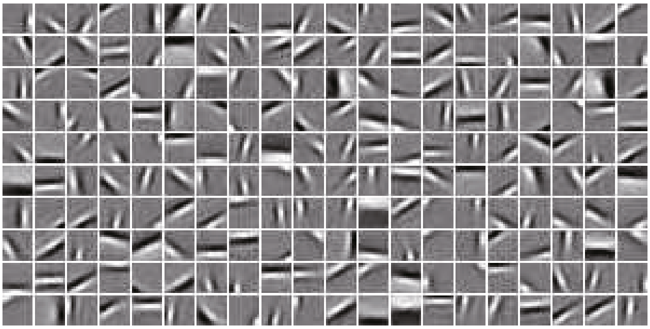


Fig. 3. Visualization of 200 first layer centroids of the hierarchical K-means

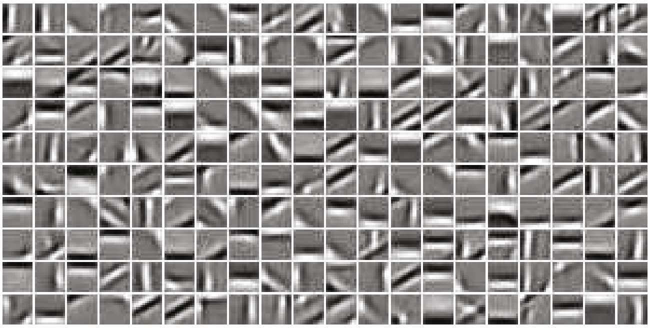


Fig. 4. Visualization of 200 second layer centroids of the hierarchical K-means.

maximum response was elicited. All measures were then taken with all angle stimuli centered at this position.

For each angle stimulus, we calculated the responses of the first layer units and second layer units sequentially. Fig. 5 shows the stimuli set with $M = 24$ together with responses of three representative second layer units. Note the similarity to Fig. 5 in [5]. And we emphasize that these units are typical in our model.

To make quantitative comparison between the simulation results and physiological results in [15], we then generated a stimuli set with $M = 12$. Five quantities about the statistics of the response profiles of the model neurons on the stimuli set were calculated and presented in Fig. 6. The physiological results and the model neurons by the sparse DBN are also presented in the figure. It is seen that the hierarchical K-means algorithm has produced similar results.

As the hierarchical K-means algorithm and the sparse DBN can produce similar results, then how about their computational efficiency? This is not a question in the computational neuroscience community but is important in engineering applications. One difficulty for such a comparison is that a common termination condition is lacked for the algorithms (notice that their final results are not the same, though qualitatively similar). Fortunately, our experiments showed that the computing time of the two algorithms differed much for producing visually

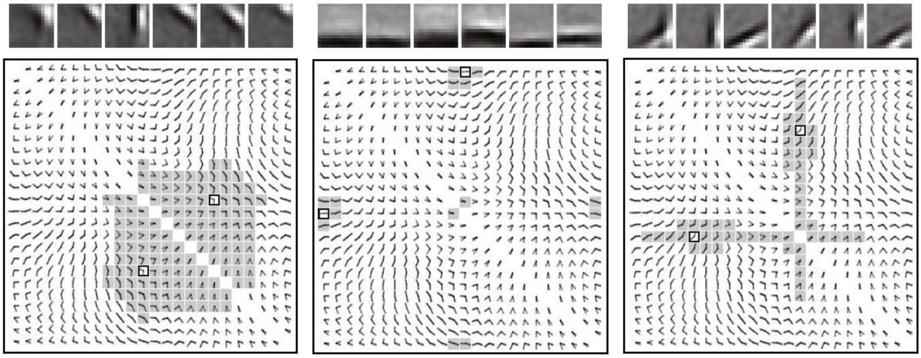


Fig. 5. Response profile of three example model V2 neurons on a set of angle stimuli. Top: the left most patch shows a model V2 neuron by taking the weighted sum of V1 simple cell receptive fields. The next five patches show the receptive fields of the model V1 simple cells that have strongest connections to this V2 neuron. Bottom: darkened patches represent stimuli to which the model V2 neuron responded strongly. A small black square indicates the overall peak response.

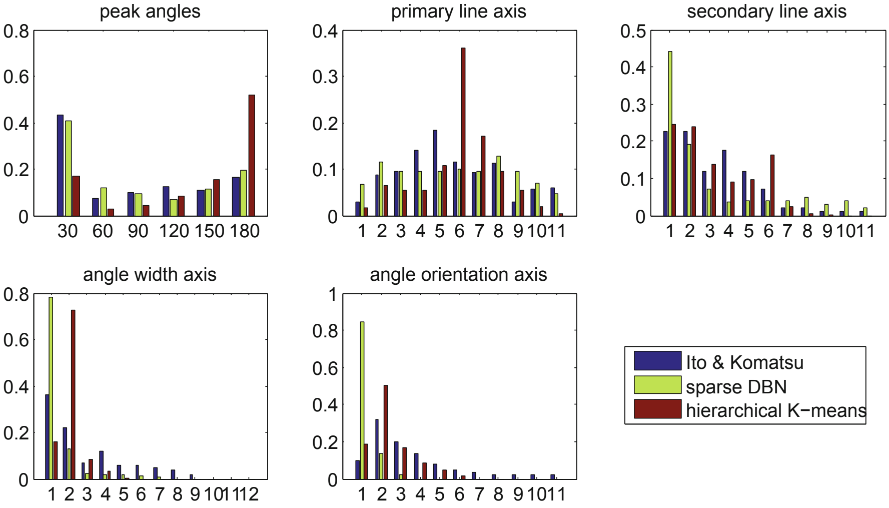


Fig. 6. Distribution of the response statistics over the angle stimuli. The five figures show respectively the distribution over (i) peak angle response, (ii) tolerance to primary line component, (iii) tolerance to secondary line component, (iv) tolerance to angle width, (v) tolerance to angle orientation. See [5, 15] for details. Best viewed in color.

similar results. Table 1 shows the computing time of the two algorithms on a computer (Intel Core i5-2320 3GHz × 4, RAM 8GB), averaged over 10 trials, for producing visually similar results to Figs. 1, 2(b), 3 and 4, respectively. For sparse DBN, p=0.02 in layer 1 and p=0.04 in layer 2. Moreover, in each layer σ decayed by a factor of 0.99 after every iteration with initial value 0.4, as suggested in [16]. Other parameters were tuned to achieve high efficiency. Learning terminated after 800 iterations for each layer and in every iteration 100,000 patches were input to the model in batches of 200. For hierarchical K-means, learning terminated after 100 iterations for each layer and in every iteration 50,000 patches were input to the model together. It is seen that learning in each layer of the hierarchical K-means algorithm is more than ten times faster than the sparse DBN.

Table 1. Comparison of the computing time in seconds

	V1	V2
sparse DBN	2536.7±21.0	2693.1±37.3
hierarchical K-means	164.3± 4.9	206.3±2.7

5 Conclusions

There are many models capable of reproducing edge-like structure of the receptive fields of V1 neurons, but few have shown to be capable of reproducing edge conjunction structure of the receptive fields of V2 neurons, except the sparse DBN. In the paper a hierarchial K-means algorithm is proposed as an alternative model for the visual area V2. The simulation results on natural images qualitatively matched physiological data recorded in monkeys. It was shown to be much more computationally efficient than the sparse DBN. A promising future direction of this research is to extend the hierarchical K-means algorithm to deep models for learning object parts for computer vision, like the convolutional DBN [17].

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grants 91120011 and 61134012, National Basic Research Program (973 Program) of China under Grant 2012CB316301, Cross-discipline Foundation and Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList).

References

1. Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. Journal of Neurophysiology 28, 229–289 (1965)

2. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381, 607–609 (1996)
3. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision Research 37(23), 3311–3325 (1997)
4. Bell, A.J., Sejnowski, T.J.: The "independent components" of natural scenes are edge filters. Vision Research 37(23), 3327–3338 (1997)
5. Lee, H., Ekanadham, C., Ng, A.: Sparse deep belief net model for visual area V2. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (eds.) Advances in Neural Information Processing Systems (NIPS), Vancouver, Canada, vol. 20 (2007)
6. Ranzato, M., Boureau, Y.L., LeCun, Y.: Sparse feature learning for deep belief networks. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (eds.) Advances in Neural Information Processing Systems (NIPS), Vancouver, Canada, vol. 20 (2007)
7. Coates, A., Lee, H., Ng, A.Y.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Ft. Lauderdale, FL (2011)
8. Saxe, A.M., Bhand, M., Mudur, R., Suresh, B., Ng, A.Y.: Unsupervised learning models of primary cortical receptive fields and receptive field plasticity. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (eds.) Advances in Neural Information Processing Systems (NIPS), vol. 24, pp. 1971–1979 (2011)
9. Karklin, Y., Lewicki, M.S.: A hierarchical bayesian model for learning nonlinear statistical regularities in nonstationary natural signals. Neural Computation 17, 397–423 (2005)
10. Karklin, Y., Lweicki, M.S.: Emergence of complex cell properties by learning to generalize in natural scenes. Nature 457, 83–86 (2009)
11. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. Nature Neuroscience 2, 1019–1025 (1999)
12. Cadieu, C., Kouh, M., Pasupathy, A., Connor, C.E., Riesenhuber, M., Poggio, T.: A model of V4 shape selectivity and invariance. Journal of Neurophysiology 98, 1733–1750 (2007)
13. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation 18, 1527–1554 (2006)
14. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation 14, 1771–1800 (2002)
15. Ito, M., Komatsu, H.: Representation of angles embedded within contour stimuli in area V2 of macaque monkeys. The Journal of Neuroscience 24(13), 3313–3324 (2004)
16. Ekanadham, C.: Sparse deep belief net models for visual area V2. Undergraduate honors thesis, Stanford University (2007)
17. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th International Conference on Machine Learning (ICML), Montreal, Canada, pp. 609–616 (2009)

Feature Selection for Unsupervised Learning

Jyoti Ranjan Adhikary and M. Narasimha Murty

Computer Science and Automation,
Indian Institute of Science, Bangalore, India
{jyotiranjanadhikary,mnm}@csa.iisc.ernet.in

Abstract. In this paper, we present a methodology for identifying best features from a large feature space. In high dimensional feature space nearest neighbor search is meaningless. In this feature space we see quality and performance issue with nearest neighbor search. Many data mining algorithms use nearest neighbor search. So instead of doing nearest neighbor search using all the features we need to select relevant features. We propose feature selection using Non-negative Matrix Factorization(NMF) and its application to nearest neighbor search.

Recent clustering algorithm based on Locally Consistent Concept Factorization(LCCF) shows better quality of document clustering by using local geometrical and discriminating structure of the data. By using our feature selection method we have shown further improvement of performance in the clustering.

Keywords: Feature selection, Non-negative matrix factorization(NMF), Locally consistent concept factorization(LCCF).

1 Introduction

The necessity to extract useful and relevant information from large datasets has led to an important need to develop computationally efficient text mining algorithms. Feature selection and dimensionality reduction techniques have become a real prerequisite for data mining applications. In both the approaches, the goal is to select a low dimensional subset of the feature space that covers most of the information of the original data.

Nearest neighbor search in high dimensional space is an interesting and important, but difficult problem. Finding the closest matching object is important for many applications. Based on that observation it has been shown in [1] that in high dimensional space, all pairs of points are almost equidistant from one another for a wide range of data distributions and distance functions. In such cases, a nearest neighbor search is unstable. So we need to reduce the dimensionality of original feature space to a reduced feature space using some feature selection approach so that in that new feature space nearest neighbor search can be meaningful.

SVM and decision tree classifiers do not use nearest neighbor search. We can select some relevant features so that we get better computational performance without significant information loss. In our work we focus on some context,

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 382–389, 2012.

such as the relationship between feature selection techniques and the resulting classification accuracy and clustering purity.

2 Background

2.1 Feature Selection

In text categorization, feature selection (FS) [9] is typically performed by sorting features according to some weighting measure and then setting up a threshold on the weights or simply specifying a number or percentage of highly scored features to be retained. Features with lower weights are discarded as having less significance. We use Embedded Feature Selection using Support vector machine (SVM) [2] and k-nearest-neighbor classifier [5]. We compare the effect of feature selection based on an SVM [4] where class labels are known with feature selection using NMF where class label information is not available.

2.2 Clustering

Non-Negative Matrix Factorization. Non-negative Matrix Factorization (NMF) [6] is a matrix factorization algorithm that focuses on the analysis of data matrices whose elements are nonnegative. The NMF consists of reduced rank non-negative factors $W \in R^{t \times k}$ and $H \in R^{k \times d}$, with $k \ll min\{t, d\}$, that approximates a given non-negative data matrix $A \in R^{t \times d}$ as $A \approx WH$. The k basis vectors $\{W_i\}_{i=1}^{k}$ can be thought of as the "building blocks" of the data, and the (d-dimensional) coefficient vector H_i describes how strongly each building block is present in the measurement vector A_i. The non-linear optimization problem underlying NMF can generally be stated as

$$\underset{W,H \geq 0}{min} f(W, H) = \frac{1}{2} \|A - WH\|_F^2 \tag{1}$$

Concept Factorization. The optimization problem underlying Concept Factorization(CF) [7] can generally be stated as

$$\underset{W,V \geq 0}{min} f(W, V) = \frac{1}{2} \|X - XWV^T\|^2 \tag{2}$$

Each column of W corresponds to one concept represented by a weighted combination of the documents in the dataset. Each row of V is a new representation of the original document vector whose each element corresponds to one cluster/concept defined by W.

Locally Consistent Concept Factorization. NMF and CF perform the factorization in the euclidean space. Locally Consistent Concept Factorization(LCCF)

[3] is a document clustering model which adds local geometry feature with CF. Using local geometry the new objective function is

$$\underset{W,V\geq 0}{min} f(W,V) = \frac{1}{2}\|X - XWV^T\|^2 + \lambda Tr(V^T LV) \tag{3}$$

where $\lambda \geq 0$ is the regularization parameter, S is defined in Eq. 6, $D_{ii} = \sum_j S_{ij}$, $L = D - S$. For non-negative data the multiplicative updating rules minimizing the above objective function are given as

$$w_{ij} \leftarrow w_{ij} \frac{(KV)_{ij}}{(KWV^T V)_{ij}} \tag{4}$$

$$v_{ij} \leftarrow v_{ij} \frac{(KW + \lambda SV)_{ij}}{(VW^T KW + \lambda DV)_{ij}} \tag{5}$$

where K is the kernel matrix. Standard kernel matrix is $X^T X$. In addition to standard kernel matrix, the normalized-cut weighted form (NCW) suggested by [8] can be used. The NCW weighting can automatically re-weight the samples which automatically balance the effect of clustering algorithms to achieve better result when dealing with unbalanced data.

3 Our Approach

3.1 Feature Selection Methods

Nearest neighbor search in high dimensional spaces affects performance and quality of text categorization. In such cases, a nearest neighbor is said to be unstable. So we need some method to reduce the dimensionality. For that purpose we use feature selection as dimensionality reduction. We use Non-negative matrix factorization for feature selection and compare it with feature selection using SVM.

SVM Feature Selection. We can use feature selection based on SVM [4] when class labels are known. So we need some feature selection method which gives selected features without class information. So we use NMF to manually get class label by clustering training samples into c clusters. Using these cluster labels c_i we can use SVM feature selection method.

SVM feature selection method selects features only from support vectors. These features are discriminative in nature because support vectors only separates classes. It ignores features which are not present in support vectors. So we need some method of feature selection which selects features from the entire feature space. We use NMF feature selection approach.

NMF Feature Selection. Feature selection using NMF uses all features and gives those features which are actually responsible for classification. NMF feature selection is given in Algorithm 1.

Algorithm 1. Feature Selection using NMF

```
Require: Given Training set $A \in R^{t \times d}$, target number of features $p$
Ensure: Selected features
Apply NMF on A to get W,H
k= number of t-dimensional basis vectors W
Select $p/k$ features from each $w_i$ having highest weight
```

We can further filter features using both NMF feature selection and SVM feature selection to get features having both discriminative and descriptive in nature.

3.2 Locally Consistent Concept Factorization with Selected Features

LCCF creates a weight matrix by using local geometric structure effectively modeled through a nearest neighbor graph on a scatter of data points. The edge weight matrix S is defined as follows:

$$S_{ij} = \begin{cases} \dfrac{x_i^T x_j}{\|x_i\| \|x_j\|} & \text{if } x_i \in N_p(x_j) \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

where $N_p(x_j)$ denotes the set of p-nearest neighbors of x_j. Constructing a graph with N vertices where each vertex corresponds to a document using k-nearest neighbors in the t-dimensional corpus may not give actual weights. So by selecting features from t-dimensional feature space and then Defining the edge weight matrix S using k-nearest neighbors in selected features will give required weight matrix. In summary, our data clustering algorithm is described in Algorithm 2.

4 Experimental Results

4.1 Data Set

Four data sets are used in our experiments. We remove features having documents frequency less than two and stop words like 'the', 'and' etc. We use the term-frequency vector to represent each document. Let $W = \{f_1, f_2, ..., f_t\}$ be the complete vocabulary set of the document corpus. We use tf-idf weighting method for global frequency of each term. The term-frequency vector A_i of document d_i is defined as

$$A_i = [a_{1i}, a_{2i}, ..., a_{ti}]^T$$

$$a_{ji} = tf_{ji}.log\left(\frac{n}{idf_j}\right)$$

where t_{ji}, idf_j, n denote the term frequency of word $f_i \in W$ in document d_i, the number of documents containing word f_i and the total number of documents. For classification we partition the datasets into 75% training samples and 25% of testing samples.

Algorithm 2. Locally Consistent Concept Factorization with selected features

```
1: Given a data set, construct the data matrix $X$ in which column $i$ represents the feature vector of data point $i$.
2: Using Algorithm 1 select relevant features and using these features construct $S$ and $D$ as given in Eq.(6)
3: Fixing $V$, update matrix $W$ to decrease the quadratic form of Eq.(3) using Eq.(4).
4: Fixing $W$, update matrix $V$ to decrease the quadratic form of Eq.(3) using Eq.(5).
5: Normalize $W$.
6: Repeat Step 3, 4 and 5 until the result converges.
7: Use matrix $V$ to determine the cluster label of each data point. More precisely, examine each row $i$ of matrix $V$. Assign data point $i$ to cluster $x$ if
       $$x = \underset{c}{argmax}(v_{ic}) \quad (7)$$
```

Table 1. Datasets

	Classic4	20NG	TDT2	Reuters
No of documents	7094	18745	9394	8067
No of classes	4	20	30	30
No of features	5896	16333	12353	18832

We use the normalized mutual information(NMI) and purity(accuracy) of clustering as our evaluation matrix [8]. The algorithms that we evaluated are Gradient Descent with Constrained Least Squares(GNMF), Concept Factorization(CF), Locally Consistent Concept Factorization(LCCF).

4.2 Results

Feature Selection. From Fig. 1 we can see that with increase in dimensionality performance of kNN classifier decreases. In Fig. 1 we compare three feature selection schemes. Feature selection using NMF always gives better performance as compared to feature selection using SVM. Because SVM feature selection method selects features only from support vectors which are discriminative in nature. It ignores features which are not present in support vectors. But NMF feature selection method selects highest weighting features from entire feature space which are descriptive in nature. By combining both SVM, NMF feature selection method we will get features with both descriptive and discriminating property.

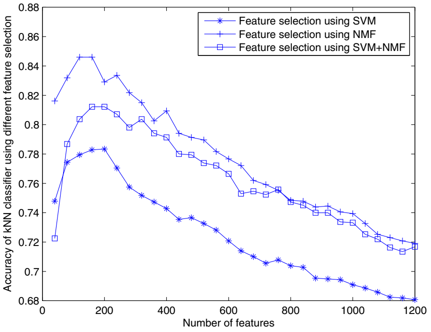


Fig. 1. Comparison of Feature selection schemes on Classic4 dataset

Classification

– kNN classifier

From Fig. 2 and 3 we can observe that nearest neighbor search is meaningful in the range of 100-200 dimensions for Classic4 dataset and 500-600 dimensions for 20Newsgroup dataset. Horizontal line in the Fig. 2 and 3 shows the accuracy of classifier in full feature space.

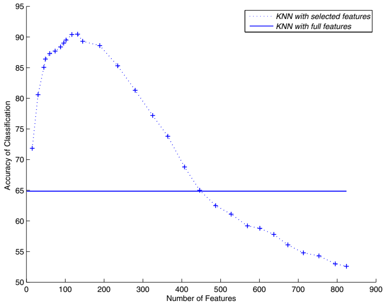


Fig. 2. kNN Classifier on Classic4

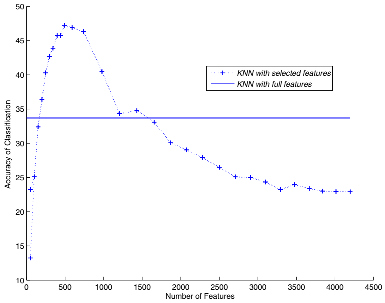


Fig. 3. kNN Classifier on 20NG

– SVM classifier

From Fig. 4 and Fig. 5 we can see an increase in accuracy with increase in number of features, but after a certain number of features it remains a constant. Horizontal lines in the Fig. 4 and 5 shows the accuracy of classifier in full feature space. We observe that after $10 - 20\%$ features, accuracy remains constant. So we need to select those useful features using feature selection approaches.

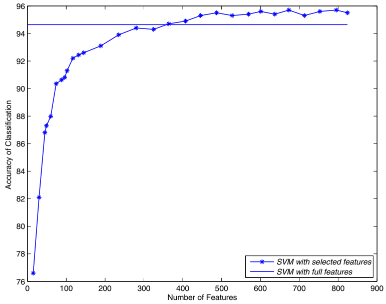


Fig. 4. SVM Classifier on Classic4

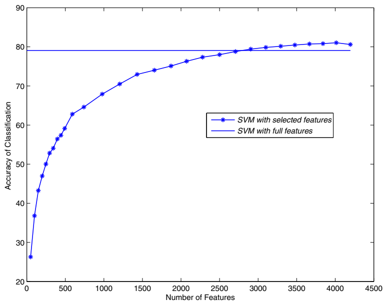


Fig. 5. SVM Classifier on 20NG

Clustering. We show performance of our proposed method Locally Consistent Concept Factorization with selected features(redLCCF) and compare it with other clustering algorithms. Tables 2 and 3 shows purity and NMI for full feature space. We observe that for less than four clusters GNMF performs well, but with increase in number of clusters our redLCCF approach gives better performance. Fig. 6 and 7 shows the purity and normalized mutual information versus the number of clusters for different algorithms on the TDT2 dataset. As can be seen, our proposed redLCCF algorithm consistently performs better than all the other algorithms.

Table 2. Purity of clustering

	Reuters	TDT2
LCCF	0.3463	0.8416
redLCCF	**0.3874**	**0.8610**
CF	0.2944	0.7312
GNMF	0.2173	0.6992

Table 3. NMI of clustering

	Reuters	TDT2
LCCF	0.3796	0.8667
redLCCF	0.**4024**	**0.8951**
CF	0.3885	0.6763
GNMF	0.3049	0.5971

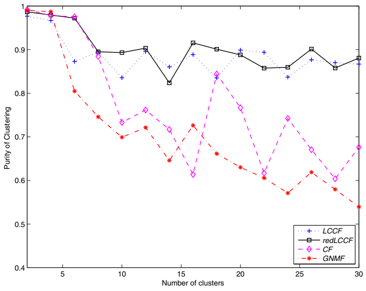


Fig. 6. Purity of clustering

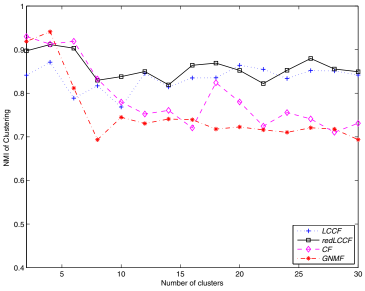


Fig. 7. NMI of clustering

5 Conclusion

By effective and efficient feature selection methods we can get almost same quality of performance with lesser computational time. From our experimental results we observe that only $10 - 20\%$ of the original features actually are responsible for the quality of performance. We use redLCCF, which is applicable to both positive and negative data values unlike NMF which works with positive data value only. Due to unconstrained nature of input data we can use kernel methods. Many tasks that use k-nearest neighbor search can be combined with our efficient feature selection approach to get better performance.

References

1. Beyer, K., Goldstein, J., Ramakrishnan, R.: Shaft, Uri.: When is "Nearest Neighbor" Meaningful? In: Int. Conf. on Database Theory (1999)
2. Christopher, J.C.B.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998)
3. Cai, D., He, X.F., Han, J.W.: Locally Consistent Concept Factorization for Document Clustering. IEEE Trans. on Knowl. and Data Eng. 23, 902–913 (2011)
4. Chapelle, O., Keerthi, S.: Multi-class Feature Selection with Support Vector Machines. In: Proceedings of the American Statistical Association (2008)
5. Cunningham, P., Delany, S.J.: K-nearest Neighbour Classifiers. Technical Report (2007)
6. Lee, D.D., Seung, H.S.: Algorithms for Non-negative Matrix Factorization. In: Advances in Neural Information Processing Systems, vol. 13, MIT Press (2001)
7. Xu, W., Gong, Y.H.: Document Clustering by Concept Factorization. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2004. ACM (2004)
8. Xu, W., Liu, X., Gong, Y.H.: Document Clustering Based on Non-negative Matrix Factorization. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR 2003. ACM (2003)
9. Yang, Y.M., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. Morgan Kaufmann Publishers (1997)

Recursive Similarity-Based Algorithm for Deep Learning

Tomasz Maszczyk[1] and Włodzisław Duch[1,2]

[1] Department of Informatics, Nicolaus Copernicus University
Grudziądzka 5, 87-100 Toruń, Poland
[2] School of Computer Engineering, Nanyang Technological University, Singapore
{tmaszczyk,wduch}@is.umk.pl
http://www.is.umk.pl

Abstract. Recursive Similarity-Based Learning algorithm (RSBL) follows the deep learning idea, exploiting similarity-based methodology to recursively generate new features. Each transformation layer is generated separately, using as inputs information from all previous layers, and as new features similarity to the k nearest neighbors scaled using Gaussian kernels. In the feature space created in this way results of various types of classifiers, including linear discrimination and distance-based methods, are significantly improved. As an illustrative example a few non-trivial benchmark datasets from the UCI Machine Learning Repository are analyzed.

Keywords: similarity-based learning, deep networks, machine learning, k nearest neighbors.

1 Introduction

Classification is one of the most important area of machine learning. Similarity-based methods (SBL, [1,2]), including many variants of the k-nearest neighbor algorithms, belong to the most popular and simplest methods used for this purpose. Although such methods have many advantages, including an easy handling of unlimited number of classes and stability of solutions against small perturbations of data, their applications are limited, because their computation time scales like $O(n^2)$ with the number of reference samples n. For large databases, especially in problems requiring real-time decisions, such "lazy approaches" relaying more on calculations performed at the time of actual classification rather than at the time of training are too slow. Training of all similarity-based methods, including kernel-based SVM approaches, also suffers from the same quadratic scaling problem. Fast methods for finding approximate neighbors can reduce this time to roughly $O(\log n)$ [3].

After decades of development simple predictive machine learning methods seem to have reached their limits. The future belongs to techniques that automatically compose many transformations, as it is done in meta-learning based on search in the model space [4,5,6], learning based on generation of novel features [7,8], and deep learning [9] approaches. The recursive SBL (RSBL) approach presented here is inspired by recent successes of the deep learning techniques. Kernel-based approaches make only

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 390–397, 2012.

one step, replacing original features with similarity-based features and performing linear discrimination in this space. Deep learning in neural networks is based on learning in new feature spaces created by adding many network layers, in essence performing recursive transformations. Instead of sequentially performing input/output transformations, RSBL version considered here systematically expands the feature space using information from all previous stages of data transformation. In this paper only transformations based on similarities to the nearest k-samples scaled by Gaussian kernel features are explored, but any other similarity measures may be used in the same way [7,8]. In essence this connects similarity-based methodology with deep learning techniques, creating higher-order k-nearest neighbors method with kernel features.

In the next two sections a distance-based and deep learning approaches are introduced. Short description of classification algorithms used here is given in the fourth section. RSBL algorithm is introduced in section 5. Illustrative examples for several datasets that require non-trivial analysis [10] are presented in section 6. Conclusions are given in the final section.

2 Similarity-Based Learning

Categorization of new points based on their distance to points in a reference (training) dataset is a simple and effective way of classification. There are many parameters and procedures that can be included in the data models M based on similarity. Such models are optimized to calculate posterior probability $p(C_i|\boldsymbol{x}; M)$ that a vector $\boldsymbol{x}$ belongs to class C_i [1,2]. Optimization includes the type of distance functions, or the type of kernel $D(\boldsymbol{x}, \boldsymbol{y})$ that should be designed depending on the problem, selection of reference instances, weighting of their influence, and other elements. The most common distance functions include:

- Minkowski's metric $D(\boldsymbol{x}, \boldsymbol{y})^\alpha = \sum_{i=1}^{d} |x_i - y_i|^\alpha$, becoming Euclidean metric for $\alpha = 2$, the city block metric for $\alpha = 1$ and the Chebychev metric for $\alpha = \infty$.
- Mahalanobis distance $D(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})'\mathbf{C}^{-1}(\boldsymbol{x} - \boldsymbol{y})}$ where $\mathbf{C}$ is the covariance matrix, taking into account scaling and rotation of data clusters.
- Cosine distance, equal to the normalized dot product $D(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x} \cdot \boldsymbol{y}/||\boldsymbol{x}||||\boldsymbol{y}||$.
- Hamming distance is used for binary features $D(\boldsymbol{x}, \boldsymbol{y}) = \#(x_i \neq y_i)/d$.
- Correlation distance is also often used:

$$D(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{d}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{d}(x_i - \overline{x})^2 \sum_{i=1}^{d}(y_i - \overline{y})^2}} \tag{1}$$

Heterogenous metric functions suitable for nominal data may be defined using conditional probabilities [1,2], but will not be used in this paper.

3 Deep Learning

Learning proceeds by reduction of information, starting from rich information at the input side and after a series of transformations creating an output sufficient for high-level decision, such as an assignment to a specific category. This information compression process can be presented as a network, a flow graph in which each node represents elementary data transformation. Flow graphs have different depth, i.e. the length of the longest path from an input to an output. Popular classification algorithms have low depth indices. For example, SVM algorithms, Radial Basis Function (RBF) networks, or the k-NN algorithms all have depth equal to two: one for the kernel/distance calculation, and one for the linear models producing outputs, or for selection of the nearest neighbors. Kernel SVM algorithms are basically linear discrimination algorithms in the space of kernel features [8], selected using the wide-margin principle. The depth of the Multilayer Perceptron (MLP) neural networks depends on the number of hidden layers and in most cases is rather small, but in deep learning it tends to be large [9].

Despite their low depth most classifiers are universal approximators, i.e. they can represent arbitrary function to a given target accuracy. In his book Bengio [9] shows examples of functions that can be represented in a simple way with deep architecture, but a shallow one may require exponentially large number of nodes in the flow graph, and may be hopelessly difficult to optimize. The most important motivation to introduce deep learning comes from signal processing by the brain. For example, the image processing by the retina, lateral geniculate nuclei and visual cortex is done in many areas, each of which extracts some features from the input, and communicates results to the next level. Each level of this feature hierarchy represents the input at a different level of abstraction, with more abstract features further up in the hierarchy, defined in terms of the lower-level ones. One may argue that this processing is in fact best approximated by a sequence of layers estimating similarity based on the lower-level similarity estimations. People organize ideas and concepts hierarchically, learning first simpler concepts and then composing them to represent more abstract ones. Engineers break-up solutions into multiple levels of abstraction and processing, using the divide-and-conquer approach at many levels. RSBL is inspired by such observations.

4 Classification Algorithms

In this section short description of classification algorithms used in our tests is presented. All of them are well known and their detailed description may be found in classic textbooks [11].

4.1 Support Vector Machines (SVM)

Support Vector Machines (SVMs) are currently the most popular method of classification and regression [12]. They require two transformations: first is based on kernels that estimate similarity $K(\boldsymbol{x}; \boldsymbol{x}_i)$ comparing the current vector $\boldsymbol{x}$ to the reference vectors $\boldsymbol{x}_i$ selected from the training set. The second transformation is based on linear discrimination, selecting from the training vectors only those reference vectors (support vectors) that are close to the decision border, with regularization term added to ensure wide-margin solutions. Depending on the choice and optimization of kernel parameters SVM is capable of creating flexible nonlinear data models that, thanks to the optimization of classification margin, offer good generalization. The best solution maximizes the minimum distance between the training vectors $\boldsymbol{x}_i$ and the points $\boldsymbol{x}$ on the decision hyperplane $\boldsymbol{w}$:

$$\max_{\boldsymbol{w},b} \min \|\boldsymbol{x}_i - \boldsymbol{x}\| \ : \ \boldsymbol{w} \cdot \boldsymbol{x} + b = 0, \ i = 1, \ldots, n \tag{2}$$

The $\boldsymbol{w}$ weight vector and the bias b are rescaled in such a way that points closest to the hyperplane $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ lie on one of the parallel hyperplanes defining the margin $\boldsymbol{w} \cdot \boldsymbol{x} + b = \pm 1$. This leads to the requirement that:

$$\forall_{\boldsymbol{x}_i} \ y_i[\boldsymbol{w} \cdot \boldsymbol{x}_i + b] \geq 1 \tag{3}$$

The width of the margin is then equal to $2/\|w\|$. The SVM algorithm is usually formulated for two classes, labeled by $y_i = \pm 1$, and presented as quadratic optimization, leading to the discriminant function of the form:

$$g(x) = \text{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x} \cdot \boldsymbol{x}_i + b\right) \tag{4}$$

where linear combination coefficients α_i are multiplied by the y_i. The dot product $\boldsymbol{x} \cdot \boldsymbol{x}_i$ is replaced by a kernel function $K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}')$ where $\phi(\boldsymbol{x})$ represents an implicit transformation of the original vectors to the new feature space. For any $\phi(\boldsymbol{x})$ vector the part orthogonal to the space spanned by $\phi(\boldsymbol{x}_i)$ does not contribute to $\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}')$ product, so it is sufficient to express $\phi(\boldsymbol{x})$ and $\boldsymbol{w}$ as a combination of $\phi(\boldsymbol{x}_i)$ vectors. The dimensionality d of the input vectors is frequently lower than the number of training patterns $d < n$, therefore $\phi(\boldsymbol{x})$ usually represents mapping into a high-dimensional space. Cover theorem [11] is frequently invoked to show advantages of increasing the dimension of the feature space. In some problems – for example the microarray data – dimensionality d may be much higher than the number of training patterns n, which is usually very small. In such cases dimensionality reduction helps to decrease noise inherent in some features. The discriminant function in the $\phi()$ space is:

$$g(\boldsymbol{x}) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b\right) \tag{5}$$

If the kernel function is linear the $\phi()$ space is simply the original space and the linear SVM discriminant function is based on cosine distances to the reference vectors $\boldsymbol{x}_i$ from the y_i class. The original features $\boldsymbol{x}_j, j = 1..d$ are replaced by new features $z_i(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{x}_i), i = 1..n$ that evaluate how close (or how similar) the vector is from the reference vectors using cosine metric. Incorporating signs in the coefficient vector $A_i = \alpha_i y_i$ the binary discriminant functions is:

$$g(\boldsymbol{x}) = \text{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i z_i(\boldsymbol{x}) + b\right) = \text{sgn}\left(\boldsymbol{A} \cdot \boldsymbol{z}(\boldsymbol{x})) + b\right) \tag{6}$$

With the proper choice of non-zero α coefficients this function projects vectors in the kernel space on a line defined by $\boldsymbol{A}$ direction, with b defining the class boundary. In non-separable case instead of using cosine distance measures it is better to use localized similarity measures, for example scaling the distance with Gaussian kernel $K_G(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\beta\|\boldsymbol{x} - \boldsymbol{x}'\|^2)$, contributing to the stability of the SVM solutions.

4.2 k-Nearest Neighbours (kNN)

This is a one of the simplest classification algorithms used in patter recognition. The k-nearest neighbors algorithms classify new objects assigning them to the most common class among the k nearest neighbors (k is typically a small positive integer). If k=1, then the object is simply assigned to the class of its nearest neighbor. Such version of kNN are often called 1NN (one nearest neighbor). The accuracy of k-nearest neighbor classification depends significantly both on the k value (which can be easy optimized using crossvalidation), and the metric used to compute distances between different examples. For continuous variables Euclidean or cosine distance is usually taken as the metric. For nominal features other measures, such as the Hamming distance or probability-dependent metrics may be used.

5 Recursive Similarity-Based Learning (RSBL)

Deep learning methodology combined with distance-based learning and Gaussian kernel features can be seen as recursive supervised algorithm to create new features, and hence used to provide optimal feature space for any classification method. Implementation of RSBL used in this paper is based on Euclidean distance and Gaussian kernel features with fixed σ=0.1, providing new feature spaces at each depth level. Classification is done either by linear SVM with fixed C=2^5, or the 1NN algorithm. The Algorithm sketched below presents steps of the RSBL; in each case parameters $k_{\max} = 20$ and $\alpha = 5$ were used.

Algorithm 1. Recursive similarity-based learning

Require: Fix the values of internal parameters: $k_{\max}$, maximum depth α, and σ (dispersion).

```
Standardize the dataset, $n$ vectors, $d$ features.
Set the initial space $\mathcal{H}^{(0)}$ using input features $x_{ij}$, $i = 1..n$ vectors and $j = 1..d$ features.
Set the current number of features $d(0) = d$.
for $m = 1$ to $\alpha$ do
    for $k = 1$ to $k_{\max}$ do
        For every training vector $\boldsymbol{x}_i$ find $k$ nearest neighbors $\boldsymbol{x}_{j,i}$ in the $\mathcal{H}^{(m-1)}$ space.
        Create $nk$ new kernel features $z_{j,i}(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{x}_{j,i}), j = 1..k; i = 1..n$ for all vectors
        using kernel functions as new features.
        Add new $nk$ features to the $\mathcal{H}^{(m-1)}$ space, creating temporary $\mathcal{H}^{(m,k)}$ space.
        Estimate error $E(m,k)$ in the $\mathcal{H}^{(m,k)}$ space on the training or validation set.
    end for
    Choose $k'$ that minimizes $E(m,k')$ error and retain $\mathcal{H}^{(m,k')}$ space as the new $\mathcal{H}^{(m)}$
    space.
end for
Build the final model in the enhanced feature space $\mathcal{H}^{(\alpha)}$.
Classify test data mapped into the enhanced space.
```

In essence the RSBL algorithm at each level of depth transforms the actual feature space into the extended feature space $\mathcal{H}^{(m)}$, discovering useful information by creating new redundant features. Note that the initial space covers d original features x_j that are available at each depth, preserving useful information that kernel SVM may discard. The final analysis in the $\mathcal{H}^{(\alpha)}$ space (and optimization of parameters at each level of RSBL algorithm, including feature selection) may be done by various machine learning methods. Once useful information is extracted many classification methods may benefit from it. The emphasis is on generation of new features using deep-learning methodology rather than optimization of learning.

In this paper only the simplest models, 1NN and linear SVM with fixed C=2^5, are used for illustration. RSBL may be presented as a constructive algorithm, with new layers representing transformations and procedures to extract and add to the overall pool more features, and a final layer analyzing the image of data in the enhanced feature space.

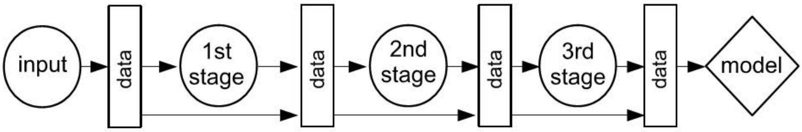


Fig. 1. RSBL method presented in graphical form for depth equal three

6 Illustrative Examples

Many sophisticated machine learning methods are introduced every year and tested on relatively trivial benchmark problems from the UCI Machine Learning Repository [13]. Most of these problems are relatively easy: simple and fast algorithms with $O(nd)$ complexity give results that are not statistically significantly worse than those obtained by the best known algorithms. Some benchmark problems are not trivial, they require complicated decision borders and may only be handled using sophisticated techniques. To distinguish dataset that should be regarded as trivial from more difficult cases simple methods with $O(nd)$ complexity have been compared with the optimized Gaussian SVM results [10].

New methods should improve results of simple low-complexity machine learning methods in non-trivial cases. Below RSBL results for a few non-trivial dataset are presented, i.e. data for which result obtained with low complexity methods are significantly worse than those obtained by kernel SVM. These datasets obtained from the UCI repository [13], are summarized in Tab. 1. In experiments 10-fold crossvalidation tests have been repeated 10 times, and the average results are collected in Tables 2-3, with accuracies and standard deviations given for each dataset.

The ORG column gives results of SVM or 1NN algorithm using the original data. RSBL(1) – RSBL(5) columns presents results in enhanced spaces at depth 1 to 5. In all cases RSBL combined with linear SVM (fixed parameters) gives results that are comparable to SVM with optimized Gaussian kernels. Additionally, increasing levels of depth provides an increase of classification accuracy (except for the *parkinsons* dataset).

The linear SVM results obtained in the RSBL enhanced feature space are almost always improved, although for this data improvement over RSBL(2) are not significant. Results of the 1NN do not improve in the enhanced space.

Table 1. Summary of datasets used in experiments

Dataset	#Vectors	#Features	#Classes
ionosphere	351	34	2
monks-problems-1	556	6	2
monks-problems-2	601	6	2
parkinsons	195	22	2
sonar	208	60	2

Table 2. 10 x 10 crossvalidation accuracy and standard deviation for RSBL combined with SVM. Additionally SVM with optimized Gaussian kernels (SVMG) results are presented for comparison.

Dataset	Method						
	ORG	RSBL(1)	RSBL(2)	RSBL(3)	RSBL(4)	RSBL(5)	SVMG
ionosphere	88.2±6.4	92.3±3.8	94.0±3.9	94.0±3.9	94.0±3.9	94.0±3.9	94.6±3.7
monks-problems-1	74.6±4.6	100±0.0	100±0.0	100±0.0	100±0.0	100±0.0	99.8±0.6
monks-problems-2	65.7±0.6	79.6±3.1	84.9±4.1	85.7±4.2	85.7±4.2	85.7±4.2	84.9±4.9
parkinsons	88.7±7.8	89.3±5.4	93.3±4.9	91.3±6.0	89.2±5.1	87.7±5.4	93.2±5.6
sonar	74.9±9.5	82.2±7.9	85.1±4.6	86.6±7.0	87.4±7.7	87.9±7.3	86.4±7.6

Table 3. 10 x 10 crossvalidation accuracy and standard deviation for RSBL combined with 1NN

Dataset	Method					
	ORG	RSBL(1)	RSBL(2)	RSBL(3)	RSBL(4)	RSBL(5)
ionosphere	87.1±5.2	87.4±4.8	87.8±4.9	87.8±4.9	87.8±4.9	87.8±4.9
monks-problems-1	100±0.0	99.9±0.1	99.4±1.2	99.3±1.2	99.4±1.1	99.4±1.0
monks-problems-2	68.8±6.2	69.2±8.7	71.6±6.2	71.6±6.2	71.6±6.2	71.8±6.2
parkinsons	93.8±5.4	92.8±6.6	91.7±6.1	91.7±6.1	91.7±6.1	91.7±6.1
sonar	85.0±5.8	85.5±6.8	86.0±6.6	87.9±6.5	87.9±6.5	87.9±6.5

7 Conclusions

The most important goal of computational intelligence is to create methods that can automatically discover the best models for a given data. There is no hope that a single method will always be the best [11], therefore such techniques like deep learning, meta-learning or feature construction methodology should be used.

RSBL algorithm introduced in this paper is focused on hierarchical generation of new distance-based and kernel-based features rather than improvement in optimization and classification algorithms. Finding interesting views on the data by systematic addition of novel features is very important because combination of such transformation-based systems should bring us significantly closer to the practical applications that automatically create the best data models for any data. Expanded feature space may benefit not only from random projections, but also from the nearest neighbor methods.

Results on several non-trivial benchmark problems shows that RSBL creates explicitly feature spaces in which linear methods reach results that are at least as good as optimized SVM with Gaussian kernels. Further improvements to the RSBL algorithm will include the use of different distance measures, fast approximate neighbors, feature selection and global optimization of the whole procedure. Applications to more challenging datasets and to the on-line learning of non-stationary data will also be considered.

Acknowledgment. This work was supported by the Nicolaus Copernicus University under research grant for young scientists no. 1141-F, and by the Polish Ministry of Education and Science through Grant no. N516 500539.

References

1. Duch, W.: Similarity based methods: a general framework for classification, approximation and association. Control and Cybernetics 29, 937–968 (2000)
2. Duch, W., Adamczak, R., Diercksen, G.: Classification, association and pattern completion using neural similarity based methods. Applied Mathematics and Computer Science 10, 101–120 (2000)
3. Arya, S., Malamatos, T., Mount, D.: Space-time tradeoffs for approximate nearest neighbor searching. Journal of the ACM 57, 1–54 (2010)
4. Duch, W., Grudziński, K.: Meta-learning via search combined with parameter optimization. In: Rutkowski, L., Kacprzyk, J. (eds.) Advances in Soft Computing, pp. 13–22. Physica Verlag, Springer, New York (2002)
5. Maszczyk, T., Grochowski, M., Duch, W.: Discovering Data Structures Using Meta-learning, Visualization and Constructive Neural Networks. In: Koronacki, J., Raś, Z.W., Wierzchoń, S.T., Kacprzyk, J. (eds.) Advances in Machine Learning II. SCI, vol. 263, pp. 467–484. Springer, Heidelberg (2010)
6. Jankowski, N., Duch, W., Grąbczewski, K. (eds.): Meta-Learning in Computational Intelligence. SCI, vol. 358. Springer, Heidelberg (2011)
7. Duch, W., Maszczyk, T.: Universal Learning Machines. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009, Part II. LNCS, vol. 5864, pp. 206–215. Springer, Heidelberg (2009)
8. Maszczyk, T., Duch, W.: Support feature machines: Support vectors are not enough. In: World Congress on Computational Intelligence, pp. 3852–3859. IEEE Press (2010)
9. Bengio, Y.: Learning deep architectures for AI. Foundations and Trends in Machine Learning 2, 1–127 (2009)
10. Duch, W., Maszczyk, T., Jankowski, N.: Make it cheap: learning with o(nd) complexity. In: IEEE World Congress on Computational Intelligence, Brisbane, Australia, pp. 132–135 (2012)
11. Duda, R.O., Hart, P.E., Stork, D.: Patter Classification. J. Wiley & Sons, New York (2001)
12. Schölkopf, B., Smola, A.: Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2001)
13. Asuncion, A., Newman, D.: UCI machine learning repository (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`

A Fast Edge-Directed Interpolation Algorithm

Qichong Tian[1], Hao Wen[1], Chenhui Zhou[2], and Wei Chen[3]

[1] Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, China
qichong.tian@gmail.com, hwen@hust.edu.cn

[2] School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, China
chzhou90@gmail.com

[3] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China
davior.chen@gmail.com

Abstract. Image interpolation is a method of obtaining a high resolution image from a low resolution image, which is applied to many image processing procedures. In order to make the interpolated image having smooth edges and make the interpolation processing fast, we propose a fast edge-directed interpolation algorithm in this paper. The proposed method consists of three steps, the determination of nonedge pixels and edge pixels, the bilinear interpolation for nonedge pixels, and the edge-adaptive interpolation for edge pixels. The experimental results show that it outperforms some existing interpolation algorithms in terms of image quality and processing speed.

Keywords: Image Interpolation, Resolution Enhancement, Edge, Image Quality.

1 Introduction

Image interpolation is a method of producing a high resolution (HR) image from the corresponding low resolution (LR) image. The method is often applied in many areas, such as image processing softwares, media players, digital cameras, high-definition TV. Some traditional interpolation algorithms, including the nearest neighbor interpolation, the bilinear interpolation and the bicubic interpolation [1], are implemented in consumer appliances. But these algorithms accordingly produce blurring or aliasing [2] around the edge area, when enlarge images.

In order to reduce the burring or aliasing and improve the visual quality, several interpolation algorithms [3]-[12], have been proposed. Allebach et al. [6] presented an edge-directed interpolation method. It generates an HR edge map from the LR image, and utilizes the linear interpolation and the correction processing to get modified pixel values. The entire process is repeated iteratively. Zhang et al. [7] proposed an edge-guided interpolation algorithm based on directional filtering and data fusion. For each

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 398–405, 2012.

unknown pixel, its neighbors are divided into two subsets. Using the subsets, two estimate pixel values are generated. Then a more robust estimate pixel value is obtained by a data fusion method. Cha et al. [8] proposed an error-amended sharp edge algorithm, which uses the bilinear interpolation firstly and corrects the interpolation error by employing the classical interpolation error theorem in an edge-adaptive way.

The method of New Edge-Direct Interpolation (NEDI) [9] is to first estimate local covariance coefficients from a LR image and then use the covariance to adapt the interpolation at an HR image based on the geometric duality between the LR covariance and the HR covariance. The NEDI method performs good visual quality of the interpolated images, but it has a very high computational complexity. In order to reduce the computational complexity, some improved NEDI algorithms have been proposed [10], [11]. However, these algorithms still have a high computational complexity compared with traditionary linear interpolations, and are not able to achieve real time image processing. To significantly reduce the computational complexity, Chen et al. [12] proposed a fast edge-oriented algorithm. The method is to first partition an image into homogeneous areas and edge areas, and then use individual interpolation algorithm respectively. In Chen's method, the detection of homogeneous area only uses two pixels around the unknown pixel and the edge-oriented adaptive interpolation for edge pixels is processing only one time, neglecting the structure of the pixels. These may accumulate the interpolation errors.

In this paper, we propose a fast edge-directed interpolation algorithm which is an improvement on the NEDI method and Chen's method. The proposed algorithm avoids the high computational complexity, and the image visual quality is better than some existing interpolation algorithms.

The rest of this paper is organized as follows. Section 2 presents a detailed description of the proposed interpolation algorithm including three steps. In section 3, the experimental results are given, in terms of image visual quality and computational complexity. Finally, Section 4 concludes the paper.

2 The Proposed Algorithm

In this section, a detail description of the proposed algorithm is given. The conceptual procedure of the proposed algorithm is presented in Fig. 1.

2.1 Determination of Nonedge Pixels and Edge Pixels

As described by Li et al. [9], an unknown pixel is declared to be an edge pixel if the local variance is above a preset threshold value. So the proposed algorithm uses the local variance estimated from the nearest neighbors to determine unknown pixels to be either nonedge pixels or edge pixels.

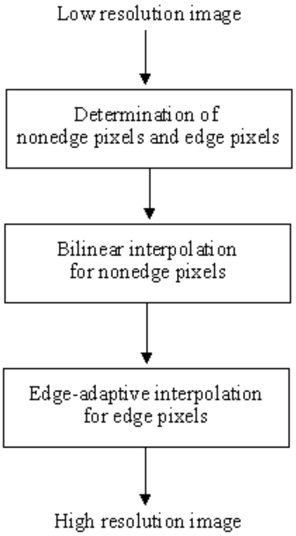

Low resolution image
Determination of nonedge pixels and edge pixels
Bilinear interpolation for nonedge pixels
Edge-adaptive interpolation for edge pixels
High resolution image

Fig. 1. The conceptual procedure of the proposed algorithm

Without the loss of generality, it assumes that the LR image **X** with size *m*×*n* directly comes from the HR image **Y** with size *2m*×*2n*, i.e., *Y(2i-1, 2j-1)=X(i, j)*. Considering the different structure of the unknown pixels, we process the pixels *Y(2i, 2j), Y(2i-1, 2j),* and *Y(2i, 2j-1)*, respectively.

The unknown pixels *Y(2i, 2j)* and the nearest neighbors are shown in Fig. 2(a). The procedure of determining nonedge pixels or edge pixels is described as follows: The letters *a, b, c,* and *d* are used to denote the luminance value of pixels *Y(2i-1, 2j-1), Y(2i-1, 2j+1), Y(2i+1, 2j-1),* and *Y(2i+1,2j+1)*, respectively. For the pixel *Y(2i, 2j)*, the local variance is equal to the variance of *a, b, c,* and *d*. If the local variance is smaller than the preset threshold, the pixel *Y(2i, 2j)* is determined to a nonedge pixel. Otherwise, the pixel *Y(2i, 2j)* is determined to an edge pixel.

The unknown pixels *Y(2i-1, 2j), Y(2i, 2j-1)* and their nearest neighbors are shown in Fig. 2(b) and Fig. 2(c). The procedure of determining nonedge pixels or edge pixels is similar as processing *Y(2i, 2j)* in Fig. 2(a).

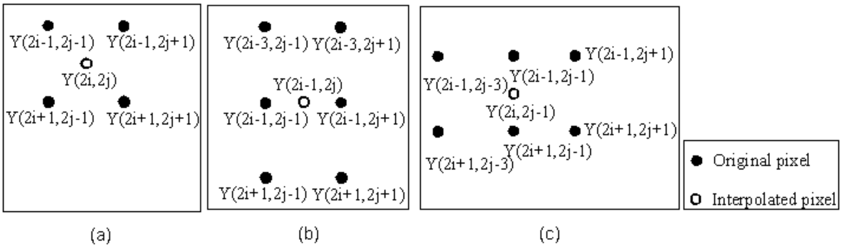


Fig. 2. The relationship between the interpolated pixels and original pixels

2.2 Bilinear Interpolation for Nonedge Pixels

The same as described in section 2.1, the nonedge pixels *Y(2i, 2j)*, *Y(2i-1, 2j)*, and *Y(2i, 2j-1)* are processed respectively. The luminance value of the pixel is equal to the average of *a, b, c,* and *d,* when the unknown nonedge pixel is *Y(2i, 2j)*. And the luminance value of the pixel is equal to the value of *(a+b+6c+6d+e+f)/16,* when the unknown nonedge pixel is *Y(2i-1, 2j)* or *Y(2i, 2j-1)*, which *a, b, c, d, e,* and *f* are luminance values of the corresponding nearest neighbor pixels.

2.3 Edge-Adaptive Interpolation for Edge Pixels

After the above process, all nonedge pixels have been defined and the edge pixels are left unknown. These edge pixels will be interpolated using an adaptive method. In this part, the algorithm scans line-by-line the image **Y**, looking for the edge pixels. The edge pixels can be classified to two types, as shown in Fig. 3 and Fig. 4. Firstly, the edge pixels *Y(2i, 2j)* are interpolated, then the edge pixels *Y(2i-1, 2j)* or *Y(2i, 2j-1)* are interpolated.

(1). In this step, the edge pixels *Y(2i, 2j)* are interpolated. As shown in Fig. 3, the values of black circle dots have been known. And the values of white square dots may, or may not, have been interpolated by the method described in section 2.2. The procedure of this step is described as follows: (a). The values of white square dots have all been interpolated by the method described in section 2.2. In this case, the algorithm computes the difference among four directions, *0°, 45°, 90°, 135°*, as shown in Fig. 3. The difference values among *0°, 45°, 90°, 135°*, are represented by *d1, d2, d3,* and *d4*, respectively. Then, the minimum value among *d1, d2, d3,* and *d4* is computed. The direction of the minimum value indicates that the edge pixel is interpolated with the known pixel on this direction. (b). The values of white square dots have not all been interpolated by the method described in section 2.2. In this case, the algorithm computes the minimum value among *d1, d2,* and *d4,* when *Y(2i-1, 2j)* or *Y(2i+1,2j)* is unknown. Or computing the minimum value among *d2, d3,* and *d4,* when *Y(2i, 2j-1)* or *Y(2i, 2j+1)* is unknown. If *Y(2i-1, 2j)* or *Y(2i+1,2j)* is an edge pixel, as well as *Y(2i, 2j-1)* or *Y(2i, 2j+1)*, the algorithm computes the minimum value between *d2* and *d4*.

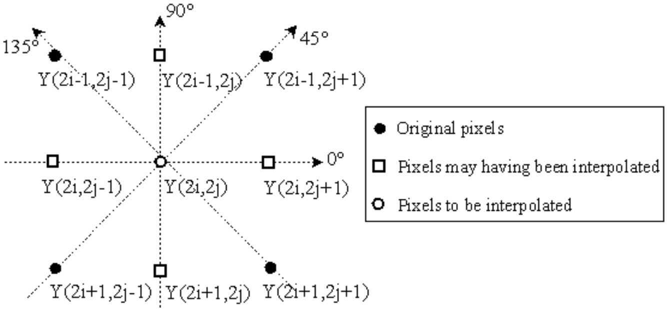


Fig. 3. The interpolation for edge pixels *Y(2i, 2j)*

(2). After the above step, the algorithm starts to process the edge pixels *Y(2i-1, 2j)* and *Y(2i, 2j-1)*. As shown in Fig. 4, the values of black circle dots and black square dots have been known. And the values of white square dots may, or may not, have been interpolated by the method described in section 2.2. The procedure of this step is similarly as the process described in the previous step.

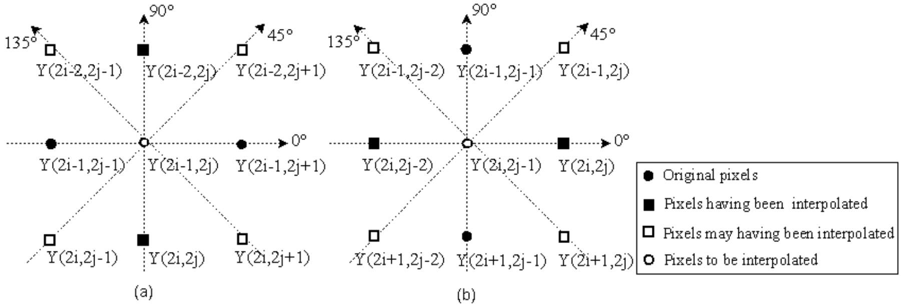


Fig. 4. The interpolation for edge pixels *Y(2i-1, 2j)* and *Y(2i, 2j-1)*

3 Experimental Results

In this section, the proposed algorithm as well as the bilinear interpolation, Chen's method [12], and the NEDI method [9] are tested. The proposed algorithm and Chen's method are implemented in Matlab by us. The NEDI Matlab code is kindly provided by its original author. While for the bilinear interpolation, the basic image processing Matlab function is used.

For the experimental needs, some synthetic and natural images representing various conditions are used, such as *Letters*(Gray scale), *Plane*(Gray scale), *Watch*(Gray scale), *Butterfly*(Color), *Lena*(Color), and *Peppers*(Color). The original test images with size 512×512 are downsampled by a factor of two along each dimension to get downsampled images with size 256×256. Then the downsampled LR images are interpolated to HR images by different interpolation methods.

3.1 Image Quality Comparison

It is generally agreed that the objective image quality assessments, such as the Peak Signal to Noise Ratio (PSNR), can not consider the visual masking effect around the arbitrarily-oriented edge and do not always provide an accurate visual quality assessment for interpolated images [7], [9]. So the subjective comparison is used to assess the visual quality of the interpolated images in this paper. For subjective comparison of different algorithms, the portions of interpolated images and original images are presented, as shown in Fig. 5. The proposed algorithm shows the most outstanding performance in preserving the smoothness of the edges, and shows the best visual quality among these interpolation methods.

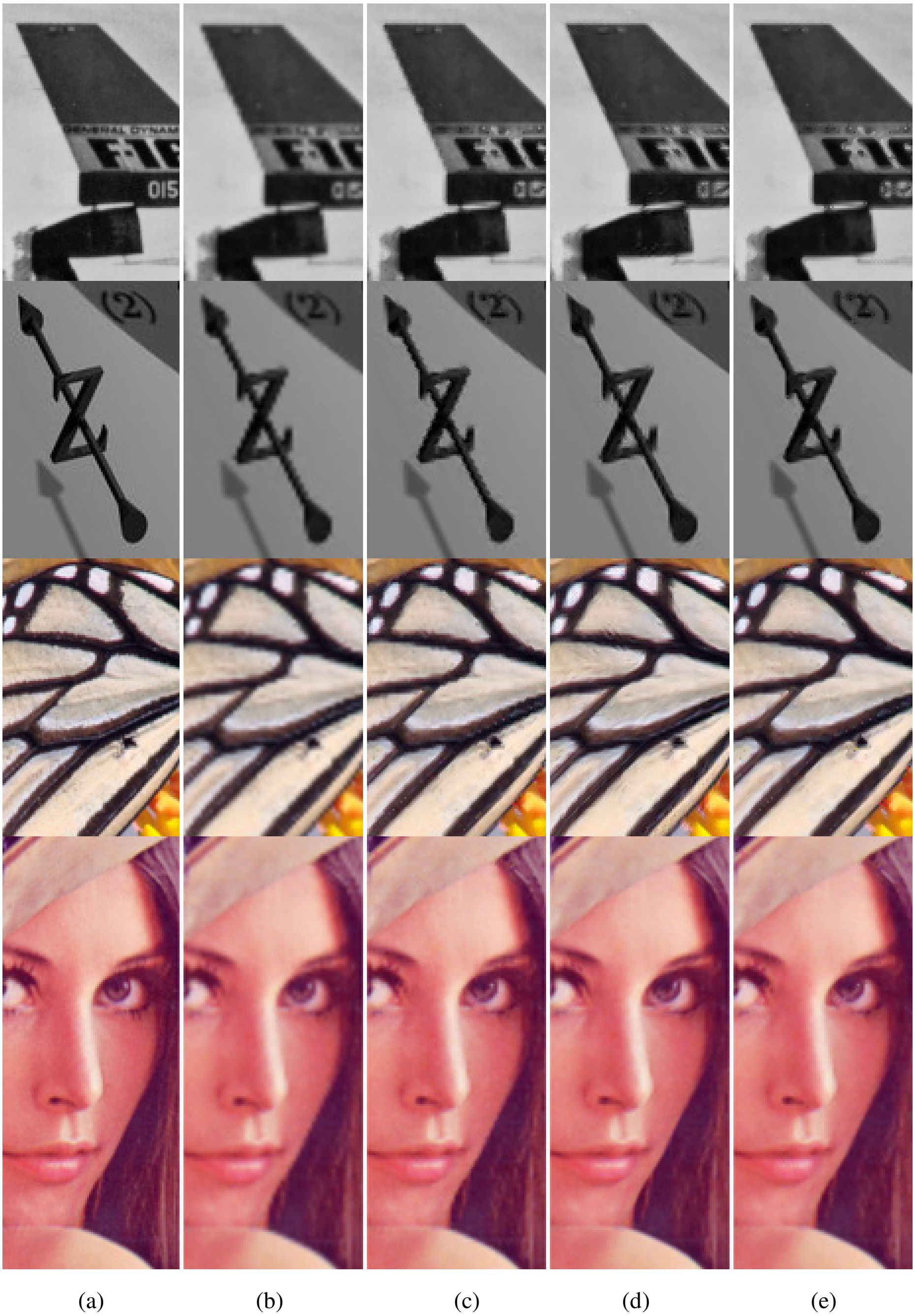


Fig. 5. Visual comparison of different interpolation algorithms:
(a) Original HR images, (b) Bilinear, (c) Chen's, (d) NEDI, (e) Proposed.

3.2 Computational Complexity Comparison

All the simulation are performed on the same platform, namely, a PC with AMD Athlon(tm) II X4 640 (3 GHz) CPU and 4GB DDR3 RAM. And all the algorithms are implemented in Matlab R2008a. So the computational time can assess the computational complexity of different methods, as shown in Table 1 and Table 2. It can draw a conclusion that the proposed algorithm has a lower computational complexity than Chen's method and the NEDI method. The low computational complexity of the proposed algorithm can make the image interpolation processing more faster.

Table 1. The computational time of gray scale images (seconds)

Images	**Bilinear**	**Chen's**	**NEDI**	**Proposed**
Letters	0.0913	0.8552	25.0636	0.2749
Plane	0.1643	1.1488	36.5660	0.5466
Watch	0.0819	0.9274	29.7018	0.5716
Average	**0.1125**	**0.9771**	**30.4438**	**0.4644**

Table 2. The computational time of color images (seconds)

Images	**Bilinear**	**Chen's**	**NEDI**	**Proposed**
Butterfly	0.2559	2.8976	110.6003	1.7239
Lena	0.2839	2.7030	102.7653	1.6144
Peppers	0.2113	2.9668	97.2152	1.6867
Average	**0.2504**	**2.8558**	**103.5269**	**1.6750**

4 Conclusion

For some applications requiring a fast image processing, such as real-time improving the resolution of videos, the visual quality and computational complexity are very important. So we propose a fast edge-directed image interpolation algorithm, which has better visual quality and lower computational complexity compared with some existing methods. The experimental results show that the proposed algorithm has good performance in the process of obtaining the HR image from the corresponding LR image. Much further work can be done on improving the image quality by using some learning algorithms or some multi-frame super-resolution algorithms.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China (Grant No. 51104157), the Fundamental Research Funds for the Central Universities (Grant No. 2011QNA30), the Ph.D. Programs Foundation of Ministry of Education of China (Grant No. 20110095120008), and the China Postdoctoral Science Foundation (Grant No.20100481181).

References

1. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice Hall, New Jersey (2002)
2. Parker, J.A., Kenyon, R.V., Troxel, D.E.: Comparison of interpolating methods for image resampling. IEEE Trans. Medical Image 2, 31–39 (1983)
3. Lehmann, T.M., Gonner, C., Spitzer, K.: Survey: interpolation methods in medical image processing. IEEE Trans. Medical Imaging 18, 1049–1075 (1999)
4. Van Ouwerkerk, J.D.: Image super-resolution survey. Image and Vision Computing 24, 1039–1052 (2006)
5. Jakhetiya, V., Kumar, A., Tiwari, A.K.: A survey on image interpolation methods. In: Processing of Second International Conference on Digital Image. SPIE, Singapore, vol. 7546, 75461T, p. 6 (2010)
6. Allebach, J., Wong, P.W.: Edge-directed interpolation. In: Proceedings of IEEE Int. Conf. on Image Processing (ICIP), Lausanne, Switzerland, pp. 707–710 (1996)
7. Zhang, L., Wu, X.: An edge-guided image interpolation algorithm via directional filtering and data fusion. IEEE Trans. Image Process. 15, 2226–2238 (2006)
8. Cha, Y., Kim, S.: The Error-Amended Sharp Edge (EASE) scheme for image zooming. IEEE Trans. Image Process. 16, 1496–1505 (2007)
9. Li, X., Orchard, M.T.: New edge-directed interpolation. IEEE Trans. Image Processing 10, 1521–1527 (2001)
10. Asuni, N., Giachetti, A.: Accuracy improvements and artifacts removal in edge based image interpolation. In: Proceedings of Int. Conf. on Computer Vision Theory and Applications (VISAPP), Funchal, Portugal, pp. 58–65 (2008)
11. Tam, W., Kok, C., Siu, W.: Modified edge-directed interpolation for images. Journal of Electronic Imaging 19, Article ID: 13011 (2010)
12. Chen, M., Huang, C., Lee, W.: A fast edge-oriented algorithm for image interpolation. Image and Vision Computing 23, 791–798 (2005)

Real-Valued Constraint Optimization with ICHEA

Anurag Sharma and Dharmendra Sharma

Faculty of Information Sciences and Engineering
University of Canberra, ACT, Australia
{Anurag.Sharma,Dharmendra.Sharma}@canberra.edu.au

Abstract. *Intelligent constraint handling evolutionary algorithm* (ICHEA) is a recently proposed variation of evolutionary algorithm (EA) that solves real-valued constraint satisfaction problems (CSPs) efficiently [20]. ICHEA has ability to extract and exploit information from constraints that guides its evolutionary search operators in contrast to traditional EAs that are 'blind' to constraints. Even its efficacy to solve CSPs it was not implemented to handle constraint optimization problems (COPs). This paper proposes an enhancement to ICHEA to solve real-valued COPs. The presented approach demonstrates very competitive results with other state-of-the-art approaches in terms of quality of solutions on well-known benchmark test problems.

Keywords: Intelligent constraint handling evolutionary algorithm (ICHEA), evolutionary algorithm (EA), constraint satisfaction problem (CSP), constraint optimization problem (COP).

1 Introduction

Evolutionary algorithm (EA) has been successful in solving many difficult NP class problems; however, it suffers from some of its inherent approaches to solve constraint problems as it does not make use of information from constraints and blindly search in the solution space using various heuristic search operators [3, 5, 16]. Characteristically, constraint problems solved by EAs use penalty based functions. A penalty function updates the fitness of chromosomes in EA. A penalty term is used in general for reward and punishment for satisfying and/or violating the constraints [4]. Use of penalty functions has been commonly reported in literature for use in constrained optimization. Two basic types of penalty functions exist; exterior penalty functions, which penalize infeasible solutions, and interior penalty functions, which penalize feasible solutions [2]. The most popular method adopted to handle constraints in EAs was taken from the mathematical programming literature – penalty functions (mostly exterior penalty functions) – where the aim is to decrease (*punish*) the fitness of infeasible solutions as to favor those feasible individuals in the selection and replacement processes. The main advantage of the use of penalty functions is their simplicity; however, their main shortcoming is that penalty factors, which determine the severity of the punishment, must be set by the user and their values are problem dependent that requires a careful fine-tuning of parameter to obtain

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 406–416, 2012.

competitive results [12, 13]. A self-adaptive penalty function based genetic algorithm (SAPF) is proposed in [21] that penalizes individuals based on ratio of total feasible and infeasible individuals present in the population. There are various forms of penalties reported in the literature, like static penalty, dynamic penalty, annealing penalty and death penalty [4].

Some other constraint handling approaches include expensive *repair* algorithms that promote the local search to transform infeasible solutions to feasible solutions because the feasible parents not necessarily produce feasible progenies [4]. In multi-objective optimization (MOO) constraints are transformed into multiple objectives. There are many established MOO algorithms like MOGA [9], VEGA [19], NSGA and NSGAII [6]. Paredis in [17] has used co-evolution strategies that utilizes *predator-prey* model to keep two populations – one population represents solutions that satisfies many constraints while other population represents those individuals whose constraint(s) is violated by lots of individuals in the first population. This strategy requires extra computational effort to find the intersection of a line with the boundary of the feasible region.

The use of domain knowledge within an EA can also be utilized to improve its performance as EAs are 'blind' to constraints. Recently, there have been few algorithms developed that move away from penalty based fitness functions to generic distance function given in Eq. (8). ICHEA [20] uses its *intermarriage* crossover operator to look for overlapping feasible regions through differentiating the boundaries of feasible regions for each constraint. This reduces the search space to obtain the solution efficiently. Cultural algorithms are also used to extract domain knowledge for its evolutionary search by using two subpopulations – population space and the belief space. Ricardo and Carlos in [18] proposed cultured differential evolution (CDE) that uses differential evolution (DE) as the population space and belief space as the information repository to store experiences of individuals for other individuals to learn. Amirjanov in [1] proposed changing domain range based genetic algorithm (CRGA) that adaptively shifts and shrinks the size of search space of the feasible region by employing feasible and infeasible solution in the population to reach the global optimum. Mezura-Montes et. al. in [14] proposed simple multi-membered evolution strategy (SMES) that uses a simple diversity mechanism by allowing infeasible solutions to remain in the population. A simple feasibility-based comparison mechanism is used to guide the process toward the feasible region of the search space. The idea is to allow the individual with the lowest amount of constraint violation and the best value of the objective function to be selected for the next population. PSO-DE proposed by [12] is another algorithm that integrates particle swarm optimization (PSO) and DE to solve real-valued COPs.

This paper is organized as follows: Section 2 describes formalization of CSPs and COPs. Section 3 revisits ICHEA introduced in [20]. Section 4 describes enhanced ICHEA that can solve COPs. Section 5 shows experimental results of ICHEA with other state-of-the-art algorithm to solve number of benchmark COPs. Section 6 discusses the outcomes of the experiments performed and section 7 concludes the paper by summarizing the results and proposing some further extensions to the research.

2 Formalization of CSPs and COPs

Constraint problems can be divided into two classes: Constrained Optimizing Problems (COPs) and constraint satisfaction problems (CSPs).The difference between these classes is that in the first an optimal solution that satisfies all the constraints should be found, while in the second class any solution as long as all the constraints are satisfied is acceptable [8]. It has been shown in [20] that ICHEA is very effective in solving real-valued CSPs, however, its ability to solve COPs was not investigated. This current work is an enhancement of ICHEA to solve real-valued COPs.

A solution to real-valued COP has two folds – search for an optimum solution that also must satisfy all the constraints. Real-valued COP can be formulated as:

$$\text{optimize } f(\vec{x}) \tag{1}$$

where COP's objective function $f(\vec{x})$ has an n-dimensional input vector $\vec{x} = \{x_1, x_2, \dots x_n\}$ that is defined in a search space S. More specifically, $\vec{x} \in \mathcal{F} \subseteq S$, where $\mathcal{F}$ being the feasible region on the search space $S \subseteq \mathbb{R}^n$. Usually, the search space S is defined as a n-dimensional rectangle in $\mathbb{R}^n$. The domain of variables is defined by their lower bounds l_i and upper bounds u_i:

$$l_i \leq x_i \leq u_i, \quad 1 \leq i \leq n \tag{2}$$

The feasible region $\mathcal{F}$ with bounds on each dimension is further restricted by a set of m additional constraints that can be given in two relational forms – equality and inequality [6, 12, 21].

$$g_i(\vec{x}) \geq 0 \qquad\qquad i = 1, \dots, k \tag{3}$$

$$h_j(\vec{x}) = 0 \quad j = k+1, \dots, m \tag{4}$$

The equality constraints $h_j(\vec{x})$ cannot be solved directly using EAs so it is converted into inequality constraints by introducing a positive tolerance value δ.

$$g_j(\vec{x}) = \delta - \left|h_j(\vec{x})\right| \geq 0 \tag{5}$$

A set of individual feasible regions $\{\mathcal{F}_1, \mathcal{F}_2, ..\mathcal{F}_m\}$ for each constraint can also be defined as:

$$\mathcal{F}_i = \{\vec{x} \in \mathcal{F} \mid g_i(\vec{x}) \geq 0, 1 \leq i \leq m, i \in Z\} \tag{6}$$

where Z is the set of integers. Many EAs uses a distance function as their fitness function to rank individuals. The distance function indicates how far a chromosome is from the feasible regions [15]. This fitness function tries to bring the chromosomes closer to the feasible region using the following function for $\forall i : \{1 \leq i \leq m\}$:

$$fitness_i(\vec{x}) = \begin{cases} g_i(\vec{x}), & if\ g_i(\vec{x}) < 0 \\ 0, & if\ g_i(\vec{x}) \geq 0 \end{cases} \tag{7}$$

$$e = \sum_{i=1}^{m} |fitness_i(\vec{x})| \tag{8}$$

The fitness function fitness_i is a measurement of *euclidean* distance of a vector $\vec{\text{x}}$ from a feasible region $\mathcal{F}_\text{i}$. The error function e is the summation of all the fitness functions. Minimizing the error value e leads toward a CSP solution where the objective function $f(\vec{x})$ is not needed. A solution to CSP is found when $e = 0$. To get a COP solution, CSP solutions are further processed to get optimum value of $\vec{\text{x}}$ that optimizes the objective function $f(\vec{x})$.

ICHEA has been demonstrated to outperform many well-known EAs to solve CSPs in [20] as it utilizes the information from constraints to guide its evolutionary search operators. The motivation behind this paper is to propose an enhancement of ICHEA to show its efficacy in solving real-valued COPs based on the test results of some benchmark problems.

3 Intelligent Constraint Handling Evolutionary Algorithm

ICHEA uses its novel search operator *intermarriage* crossover that uses information from constraints rather than blindly searching for the solution. In this crossover both parents belong to different feasible regions $\mathcal{F}_i$ and $\mathcal{F}_j$where $i \neq j$. It is also possible that a parent does not belong to any of the feasible regions $S - \mathcal{F}$. The generated offspring contains genes from both parents. The purpose is to make a "generic" offspring that tries to satisfy more than one constraint because its parents are from two different feasible regions. The algorithm favors those offspring which satisfy more constraints by utilizing Deb's ranking scheme based on feasibility [6] to rank the population where the population is first sorted according to number of satisfied constraints in decreasing order then by fitness value given in Eq. (8) in increasing order.

3.1 Intermarriage Crossover for Real-Valued CSPs

In *intermarriage* crossover, two parents generate two offspring. This is a dual process where both parents move closer to each other one at a time and their new positions are considered as two new offspring. An offspring from two parents through intermarriage is defined in a search space as a constant multiple of difference of two parent vectors as shown in Eq. (9). Initially offspring O_1 is placed at position $(P_2 - P_1)/r$ where r is a coefficient in the range within $(0,1)$ which is 0.75 if both parents satisfy at least one different constraint and r is 0.1 if both parents satisfy all same constraints. Then O_1 moves iteratively closer to parent P_1 until it also satisfies the constraint(s) that P_1 satisfies and similarly offspring O_2 is designated. The iterative move can be captured as:

$$O_1 = r^i(P_2 - P_1) \tag{9}$$

Variable i gets incremented from 1 to a threshold value Tin the sequence $\langle 1,2,\ldots,T\rangle$. We have used $T = 5$ for our experiments. So using the Eq. (9) the i value is incremented by 1 until the offspring finds an acceptable place or a threshold value T is reached. This causes two selected vectors (parents) of different constraint satisfaction sets to come closer (offspring) towards constraint boundary because the solution space lies in the overlapping boundary region. Favoring points for *intermarriage* that satisfy more constraints, results in finding solution space quickly [20].

This *intermarriage* crossover tends to converge quickly resulting in low diversity of the population. To avoid this early convergence, the concept of multi-parent crossover has been incorporated where rather than picking most desirable parents from the population, new parents are generate on the vertices of a hyper rectangle that encloses a parent. This hyper rectangle is dynamically created from the locations of two chosen parents P_i and P_j for crossover. To make a hyper rectangle around each parent the following steps are being followed:

- Determine the distance from P_j to P_i $\Delta P_{j,i}$ which is then multiplied by $dimMatrix$. $dimMatrix$ is a square diagonal matrix of size n which is the total dimensions of the search space. The diagonal entries are only ± 1 as shown below. $dimMatrix$ produces 2^n possible combinations of matrices that are used to generate set of all 2^n vertices $P_{dimMatrix}$ of the hyper rectangle where only maximum of up to 2 vertices are chosen randomly. An instance i of $dimMatrix$ namely $dimMatrix_i$ is chosen to create a parent $P_{dimMatrix_i}$ which represents the i^{th} vertex of the hyper rectangle. Matrix multiplication of $dimMatrix_i$ and $\Delta P_{j,i}$ gives the distance from new parent $P_{dimMatrix_i}$ to P_i denoted by $\Delta P_{dimMatrix_i,i}$.

$$dimMatrix = \begin{bmatrix} \pm 1 & 0 & \cdots & 0 & 0 \\ 0 & \pm 1 & & 0 & 0 \\ \vdots & & \ddots & \vdots & \\ 0 & 0 & \cdots & \pm 1 & 0 \\ 0 & 0 & & 0 & \pm 1 \end{bmatrix}$$

$$\Delta P_{j,i} = (P_j - P_i)$$

$$\Delta P_{dimMatrix_i,i} =$$
$$\Delta P_{j,i} \times dimMatrix_i$$

- Add vector P_i to the distance vector $\Delta P_{dimMatrix_i,i}$ to get parent $P_{dimMatrix_i}$:

$$P_{dimMatrix_i} = P_{\mathrm{i}} + \Delta P_{dimMatrix_i,i} \tag{10}$$

Parent P_{i} goes through intermarriage crossover with each of these parents and then only the best offspring is selected to go into the offspring pool. This same process is repeated for other parent P_{j}.

3.2 ICHEA Algorithm

ICHEA is a variation of EA introduced in that adds constraint handling features to the standard GAs. The pseudocode can be given as:

```
chromosomes  = initializeChromosomes();
for each generation
  parents = NoveltyTournamentSelection();
  offspring = interMarriageCrossover(parents);
  Mutation(offspring);
  chromosomes = chromosomes + offspring;
  SortAndReplace();
  CheckTerminationCriteria();
End for loop;
```

The detailed description of the algorithm can be found in [20]:

4 ICHEA for Constraint Optimization Problems

ICHEA introduced in [20] is limited to works for CSPs only. We have enhanced the algorithm as below to improve the solutions of the COPs as well.

4.1 Parallel Processing for CSP and COP

The foundation of ICHEA lies in acknowledging the information from the set of feasible regions $\mathcal{F}$ that guides its evolutionary search to solve CSPs effectively. To enhance its capability in solving COPs a formative approach is taken where ICHEA runs two processes in parallel – one to solve CSP and another to optimize CSP solutions. The parallel process starts by dividing the whole population pop into 2 parts. First part pop_{COP} keeps the CSP solutions that are required for optimization and the second part pop_{CSP} keeps the $good$ infeasible solutions that are processed to get CSP solutions. The ratio of pop_{COP}: pop_{CSP} is fine-tuned to 1:4 for our experiments.

pop_{CSP} is divided into equal sized m slots where slot i is allocated for individuals that violate i constraints. If there are no individuals with i violations then its allocated space is evenly distributed to other slots. This is done to keep diverse population of partially feasible solutions as [12] have observed that only keeping individuals with lower degree of constraint violations might cause the population to be trapped in a local optimum. Let pop_{CSP_i} indicate the population of individuals that violate i constraints so the total population pop_{CSP} is:

$$pop_{CSP} = \sum_{i=1}^{m} pop_{CSP_i}$$

Then pop_{CSP_i} is sorted according to the fitness and the best $|pop_{CSP}|/m$ is selected for subpopulation pop_{CSP_i}.

$$\therefore \max\left(\left|pop_{CSP_i}\right|\right) = |pop_{CSP}|/m$$

If after allocation, k slots have $\left|pop_{CSP_i}\right| < |pop_{CSP}|/m$, then unallocated population of individuals $pop_{unalloc}$ is:

$$pop_{unalloc} = \sum_{i=1}^{m} \begin{cases} |pop_{CSP}|/m - \left|pop_{CSP_i}\right|, & if \left|pop_{CSP_i}\right| < |pop_{CSP}|/m \\ 0 & , otherwise \end{cases}$$

This unallocated population $pop_{unalloc}$ needs to be allocated evenly in the slots that have $\left|pop_{CSP_i}\right| > |pop_{CSP}|/m$.

4.2 Search Focus towards Best So Far Individual

Intermarriage crossover guides the evolutionary search to focus on feasible regions. In addition to normal *intermarriage* crossover the same parents undergo intermarriage crossover with a neighbor of current best solution to guide the search focus towards best so far individual. This is similar to PSO approach [7] where all swarm particles tend to move towards better positions nearby the best position that leads to optimum solution [7, 10]. This helps in exploring promising solution in a nearby region of the current best solution. If the *intermarriage* crossover operator is denoted by ⊗ then the *intermarriage* crossover initiated by parents P_i and P_j involves the following steps:

1. $P_i \otimes P_j$
2. $P_i \otimes P_{dimMatrix_i}\ where\ \{P_{dimMatrix_i} \in P_{dimMatrix} | i\ \in randSet \wedge$ $|randSet| = 2\}\ where\ randSet =\ \{\exists i\colon 1 \leq i \leq |P_{dimMatrix}|\}$
3. $P_{neighbor_j} = \sigma(P_j + P_{best})$ where $\sigma \in (0.0,1.0)$
4. $P_i \otimes P_{neighbor_j}$

The step (1) is just a normal intermarriage crossover between P_i and P_j followed by step (2) that is an intermarriage crossover between a parent P_i and aforementioned newly created parents on the vertices of the hyper-rectangle $\forall P_{dimMatrix_i}$ (see Section 3.1) so that exploration is not limited to the selected population only. Step (3) determines the common neighbor $P_{neighbor_j}$ of parent P_j and the current best chromosome P_{best} using a randomly generated coefficient σ in the range of (0.0, 1.0). Finally intermarriage crossover happens between P_i and $P_{neighbor_j}$ in step (4) which is inspired from PSO to search near by the current best solution. These four steps are specifically used to find the COP solution.

5 Experiment

To validate the efficacy of ICHEA, 11 benchmark problems from COP domain [11, 12, 15] have been selected. All test problems are mathematical functions of various types like quadratic, linear, nonlinear and trigonometric. ICHEA has been compared against five state-of-the-art approaches briefly mentioned in the section 1: CRGA [1], SAPF [21], PSO-DE [12], CDE [18] and SMES [14]. No parallel processing or distributed environment is used for the experiments.

Table 1. Parameter Settings

Parameters	ICHEA
Population size	100
Maximum generations	1.0E3
Maximum evaluations	1.0E6
Mutation rate	0.1
Crossover rate	1.0

An average of 10 successive runs for ICHEA is taken into account to demonstrate its solution quality against published results of other algorithms mentioned above. Table 1 shows the parameter settings used for all test problems. Generally, ICHEA is able to find a solution close to optimal solution in a few generations but it is allowed to run full 1.0E3 generations to try to obtain best possible solutions. For example best solutions for problem G12, G08, G11 and G01 are obtained in 10, 12, 28, 234 generations with 9.1E3, 1.1E4, 2.4E4, 2.4E5 evaluations respectively. The positive tolerance value δ for problem G03 and G11 is 1.0E-3 and 1.0E-5 respectively.

Table 2 shows the statistical summary of the results for all the tested problems showing best, median, mean and worst solutions obtained with their corresponding standard deviations (SD). Table 3 – Table 5 show the same results compared with

Table 2. Experimental results of ICHEA on 11 benchmark functions

Functions	Best	Median	Mean	Worst	SD
G01	**-15.00000**	-15.00000	**-15.00000**	**-15.00000**	5.4E-07
G02	-0.803036	-0.784636	**-0.768525**	-0.743884	2.3E-02
G03	-1.00497	-1.00483	**-1.00476**	**-1.00483**	1.1E-04
G04	**-30665.539**	-30665.539	-30665.537	-30665.530	3.2E-03
G06	**-6961.814**	-6961.813	**-6961.814**	**-6961.814**	1.85E-05
G07	24.6149	24.9502	25.7139	27.2705	1.0E+00
G08	**-0.095825**	-0.095825	**-0.095825**	**-0.095825**	2.3E-07
G09	680.645	680.742	680.774	680.995	1.1E-01
G10	7128.097	7165.736	7196.508	7297.964	5.8E+01
G11	**0.7500**	0.7500	**0.7500**	**0.7500**	3.2E-05
G12	**-1.00000**	-1.00000	**-1.00000**	**-1.00000**	1.2E-06

Table 3. Comparison of best solutions of ICHEA with five other state-of-the-art algorithms

Functions	ICHEA	CRGA	SAPF	PSO-DE	CDE	SMES
G01	**-15.00000**	-14.9977	-15.000	-15.000000	-15.000000	-15.000
G02	-0.803036	-0.802959	-0.803202	-0.8036145	-0.803619	-0.803601
G03	-1.00497	-0.9997	-1.000	-1.0050100	-0.995413	-1.000
G04	**-30665.539**	-30665.520	-30665.401	-30665.539	-30665.539	-30665.539
G06	**-6961.814**	-6956.251	-6961.046	-6961.8139	-6961.8139	-6961.814
G07	24.6149	24.882	24.838	24.30621	24.30621	24.327
G08	**-0.095825**	-0.095825	-0.095825	-0.095826	-0.095825	-0.095825
G09	680.645	680.726	680.773	680.6301	680.6301	680.632
G10	7128.097	7114.743	7069.981	7049.248	7049.248	7051.903
G11	**0.7500**	0.750	0.749	0.749999	0.7499	0.75
G12	**-1.00000**	-1.000000	-1.000000	-1.000000	-1.000000	-1.000
Top ranked	7/11	3/11	4/11	**11/11**	**9/11**	**11/11**

other algorithms based on best, mean and worst solutions respectively. The results in bold indicate the optimum solutions or one of the best amongst all the algorithms. ICHEA is able to reach global optimum for problems G01, G04, G06, G08, G11 and G12 while problems solutions for G02, G03, G09 is very close to optimum solutions. For problems G10 very good solutions are not observed within the limited generations. This demonstrates the competitiveness of ICHEA with other algorithms.

We have also taken the count of final results that are ranked in top half, achieved by all the algorithms. The last rows of Table 3 – Table 5 shows the count of top ranked final results where PSO-DE, SMES and ICHEA are found to be best 3 out of 6 algorithms for getting good mean and worst solutions and PSO-DE, SMES, CDE and ICHEA are best 4 out of 6 algorithms for reaching towards optimum solution; however, according to “no-free-lunch” theorem no algorithm is the best for all classes of problems [22]. PSO-DE is able to demonstrate very impressive results for benchmark COPs but it is not able to perform well for CSPs as demonstrated in [20] where ICHEA outperforms it in terms of *success rate* and efficiency.

Table 4. Comparison of mean solutions of ICHEA with five other state-of-the-art algorithms

Functions	ICHEA	CRGA	SAPF	PSO-DE	CDE	SMES
G01	**-15.00000**	-14.9850	-14.552	-15.000000	-14.999996	-15.000
G02	**-0.768525**	-0.764494	-0.755798	-0.756678	-0.724886	-0.785238
G03	**-1.00476**	-0.9972	-0.964	-1.0050100	-0.788635	-1.000
G04	-30665.537	-30664.398	-30665.922	-30665.539	-30665.539	-30665.539
G06	**-6961.814**	-6740.288	-6953.061	-6961.8139	-6961.8139	-6961.284
G07	25.7139	25.746	27.328	24.30621	24.30621	24.475
G08	**-0.095825**	-0.095819	-0.095635	-0.0958259	-0.095825	-0.095825
G09	680.774	681.347	681.246	680.6301	680.6301	680.643
G10	7196.508	8785.149	7238.964	7049.248	7049.248	7253.047
G11	**0.7500**	0.752	0.751	0.749999	0.757995	0.75
G12	**-1.00000**	-1.000000	-0.99994	-1.000000	-1.000000	-1.000
Top ranked	**8/11**	2/11	1/11	**10/11**	6/11	**9/11**

Table 5. Comparison of worst solutions of ICHEA with five other state-of-the-art algorithms

Functions	ICHEA	CRGA	SAPF	PSO-DE	CDE	SMES
G01	**-15.00000**	-14.9467	-13.097	-15.000000	-14.999993	-15.000
G02	-0.743884	-0.722109	-0.745712	-0.6367995	-0.590908	-0.751322
G03	**-1.00483**	-0.9931	-0.887	-1.0050100	-0.639920	-1.000
G04	-30665.530	-30660.313	-30656.471	-30665.539	-30665.539	-30665.539
G06	**-6961.814**	-6077.123	-6943.304	-6961.8139	-6961.8139	-6952.482
G07	27.2705	27.381	33.095	24.3062	24.3062	24.843
G08	**-0.095825**	-0.095808	-0.092697	-0.0958259	-0.095825	-0.095825
G09	680.995	682.965	682.081	680.6301	680.6301	680.719
G10	7297.964	10826.09	7489.406	7049.248	7049.249	7638.366
G11	**0.7500**	0.757	0.757	0.750001	0.796455	0.75
G12	**-1.00000**	-1.000000	-0.999548	-1.000000	-1.000000	-1.000
Top ranked	**8/11**	1/11	1/11	**10/11**	7/11	**9/11**

6 Discussion

ICHEA was initially introduced to solve real-valued CSP solutions only where it was able to outperform many other EAs in terms of success rate and efficiency [20]. In this paper ICHEA has been enhanced to solve COPs as well. The comparative test results on benchmark COPs are very promising and competitive with other state-of-the-art algorithms. ICHEA is a problem independent formulation where consistent results have been observed for all the test problems using common parameters.

Introduction of ICHEA in [20] demonstrated that extracting information from constraints can produce very good solutions efficiently. Hence the basic structure of ICHEA has been kept intact while enhancing it to employ constraint optimization tasks. The current form of ICHEA is still problem independent where addition of parallel processing simultaneously deals with constraint satisfaction and optimization tasks. Intermarriage crossover has been adjusted to search for an optimum solution that still utilizes information from the constraints.

7 Conclusion

ICHEA introduced in [20] has been demonstrated to outperform many well-known EAs including PSO-DE to solve benchmark CSPs. ICHEA has been enhanced in this paper without losing its integrity to solve real-valued COPs which has shown very competitive results. This new ICHEA runs in two parallel processes – one for CSP and another for COP. The CSP process searches feasible regions to make a population of feasible solutions while COP process tries to optimize the solutions using the whole population. The main idea remains the information extraction from constraints that reduces the search space to promising regions only. Currently ICHEA is restricted to solve only real-valued CSP and COP but it has all the potential to be extended to work for discrete constraints problems as it relies on extracting information from constraints. The future work also involves applying ICHEA for dynamic CSPs and COPs.

References

1. Amirjanov, A.: The development of a changing range genetic algorithm. Computer Methods in Applied Mechanics and Engineering 195(19-22), 2495–2508 (2006)
2. Back, T., et al. (eds.): Handbook of Evolutionary Computation. IOP Publishing Ltd. (1997)
3. Brailsford, S.: Constraint satisfaction problems: Algorithms and applications. European Journal of Operational Research 119(3), 557–581 (1999)
4. Coello Coello, C.A.: Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer Methods in Applied Mechanics and Engineering 191(11-12), 1245–1287 (2002)
5. Craenen, B.G.W.: Solving constraint satisfaction problems with evolutionary algorithms. Phd Dissertation, Vrije Universiteit (2005)
6. Deb, K., et al.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
7. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS 1995, pp. 39–43 (1995)
8. Eiben, A.E.: Evolutionary algorithms and constraint satisfaction: definitions, survey, methodology, and research directions. Presented at the (2001)
9. Fonseca, C.M., Fleming, P.J.: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Proceedings of the 5th International Conference on Genetic Algorithms, pp. 416–423. Morgan Kaufmann Publishers Inc., San Francisco (1993)
10. Onwubolu, G.C., Sharma, A.: Particle Swarm Optimization for the assignment of facilities to locations. In: New Optimization Techniques in Engineering. Springer (2004)

11. Liang, J.J., et al.: Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-parameter Optimization. Nanyang Technological University, Singapore (2006)
12. Liu, H., et al.: Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl. Soft Comput., 629–640 (2010)
13. Mezura-montes, E., Coello, C.A.C.: A Survey of Constraint-Handling Techniques Based on Evolutionary Multiobjective Optimization. Departamento de Computación, Evolutionary Computation Group at CINVESTAV (2006)
14. Mezura-Montes, E., Coello, C.A.C.: A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Transactions on Evolutionary Computation 9(1), 1–17 (2005)
15. Michalewicz, Z., Schoenauer, M.: Evolutionary algorithms for constrained parameter optimization problems. Evolutionary Computation 4(1), 1–32 (1996)
16. Müller, T.: Constraint-based Timetabling. PhD Dissertation, Charles University (2005)
17. Paredis, J.: Co-evolutionary Constraint Satisfaction. In: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature, pp. 46–55. Springer (1994)
18. Becerra, R.L., Coello Coello, C.A.: Cultured differential evolution for constrained optimization. In: Computer Methods in Applied Mechanics and Engineering, vol. 195(33-36), pp. 4303–4322 (2006)
19. Schaffer, J.D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In: Proceedings of the 1st International Conference on Genetic Algorithms, pp. 93–100. Erlbaum Associates Inc. (1985)
20. Sharma, A., Sharma, D.: ICHEA – A Constraint Guided Search for Improving Evolutionary Algorithms. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part I. LNCS, vol. 7663, pp. 269–279. Springer, Heidelberg (2012)
21. Tessema, B., Yen, G.G.: A Self Adaptive Penalty Function Based Algorithm for Constrained Optimization. In: IEEE Congress on Evolutionary Computation, CEC 2006, pp. 246–253. IEEE (2006)
22. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)

Modeling Post-training Memory Transfer in Cerebellar Motor Learning*

Tadashi Yamazaki[1,2] and Soichi Nagao[2]

[1] Graduate School of Informatics and Engineering,
The University of Electro-Communications
[2] Laboratory for Motor Learning Control, RIKEN Brain Science Institute
iconip12@neuralgorithm.org

Abstract. The cerebellum has two distinct memory sites. A single session of behavioral training forms short-term memory in the cerebellar cortex, and by repeating the training, long-term memory is formed in the cerebellar or vestibular nuclei, as if the memory is transferred from the cortex to the nuclei. We propose a simple network model of the cerebellum for the formation and transfer of motor memory. We assume a Hebbian rule with a postsynaptic gating mechanism for synaptic plasticity in the nuclei. We carry out computer simulation of gain adaptation of vestibulo-ocular reflex (VOR) and demonstrate robust memory transfer: the synaptic weight in the nuclei does not diverge to infinity. We suggest that memory transfer occurs mainly after training, not during training, and that spontaneous activity of Purkinje cells after training is necessary for memory transfer.

Keywords: Cerebellum, Motor learning, Memory consolidation, Post-training period.

1 Introduction

There is a long-lasting debate on the location of motor memory in the cerebellum: the cerebellar cortex versus the cerebellar/vestibular nuclei in the brain stem [1–4]. Accumulating experimental evidence indicates that short-term memory, which is acquired by a single session of behavioral training and disappears within 24 hours, is formed at parallel fiber (PF)-Purkinje cell synapses in the cerebellar cortex, whereas long-term memory, which is acquired by repeating the session and persists for days and weeks, is formed at mossy fiber (MF)-nuclear cell synapses [5–8]. The long-term memory is formed not during training but after training, during which short-term memory decays spontaneously, as if the short-term memory in the cerebellar cortex is transferred to the nuclei and consolidated as long-term memory [9–11]. The neural mechanism of this transsynaptic memory formation, or called "memory transfer", during *post-training periods* remains unknown.

* This study was supported by KAKENHI (20700301) and an internal startup fund of the University of Electro-Communications.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 417–424, 2012.

We propose a theoretical model of post-training memory transfer in the cerebellum. We assume that PF-Purkinje cell synapses are modified by conventional long-term depression (LTD) and long-term potentiation (LTP), in which plastic change is induced by the instruction signal fed by climbing fibers (CFs). On the other hand, we assume that MF-nuclear cell synapses are updated by a Hebbian rule with a postsynaptic gating factor. By carrying out computer simulations of gain adaptation of vestibulo-ocular reflex (VOR), we demonstrate that the model exhibits memory transfer during post-training periods. We also demonstrate that spontaneous activity of Purkinje cells after training is necessary for memory transfer.

2 Model of Cerebellar Motor Learning

Figure 2 illustrates our simple network model of the cerebellum for VOR gain adaptation. Mossy fibers transmit information on head rotation to the vestibular nucleus and granule cells. Granule cells send axons called PFs to Purkinje cells, which in turn inhibit the vestibular nucleus. Granule cells also excite molecular layer interneurons such as basket and stellate cells, which in turn inhibit Purkinje cells. Climbing fibers send retinal slip information to Purkinje cells as an instruction signal to induce plastic change at PF-Purkinje cell synapses. We omit Golgi cells for the sake of simplicity.

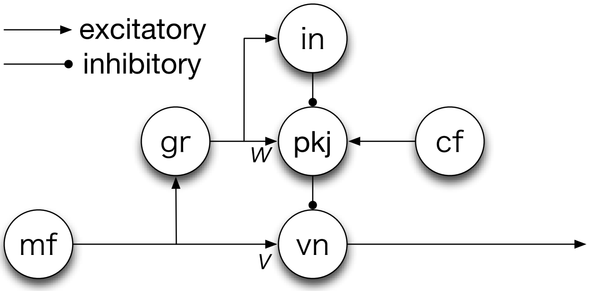


Fig. 1. Our simple network model of the cerebellum. Mossy fibers and climbing fibers provide inputs, whereas the vestibular nucleus issues the final output. Mossy fibers excite granule cells and the vestibular nucleus. Granule cells excite Purkinje cells and inhibitory interneurons that inhibit Purkinje cells. Purkinje cells inhibit the vestibular nucleus. Climbing fibers provide teacher or error signals to Purkinje cells, which induce learning of synaptic weights between granule cells and Purkinje cells. w and v are synaptic weights between granule cells and Purkinje cells, and between mossy fibers and vestibular nucleus, respectively. Abbreviations: cf, climbing fibers; gr, granule cells; in, inhibitory interneurons; mf, mossy fibers; pkj, Purkinje cells; vn, vestibular nucleus.

We calculate the activity of the neurons in our model as follows:

$$\mathrm{gr} = \mathrm{mf} \tag{1}$$

$$\mathrm{pkj} = w \cdot \mathrm{gr} \tag{2}$$

$$\mathrm{in} = \mathrm{gr} \tag{3}$$

$$\mathrm{vn} = v \cdot \mathrm{mf} - (\mathrm{pkj} - \mathrm{in})\,, \tag{4}$$

where mf, gr, pkj, in, and vn denote the activity of MFs, granule cells, Purkinje cells, molecular-layer interneurons, and vestibular nucleus, respectively, and w and v represent synaptic weights between a PF and a Purkinje cell and between an MF and a vestibular nucleus, respectively.

The value of w is initialized to $w(0)$ and updated by two different rules either during training or after training:

$$\text{Training: } \tau_{w_{\text{induction}}} \dot{w} = -\left(w - w(0)\right) + \mathrm{cf} \tag{5}$$

$$\text{Post-training: } \tau_{w_{\text{recovery}}} \dot{w} = -\left(w - w(0)\right) \tag{6}$$

where the dot notation of a variable (e.g., $\dot{w}$) denotes the time derivative of the variable (i.e., dw/dt). cf is the activity of CFs as the instruction signal for learning. We do not model the process of plasticity [12] explicitly. Rather, we consider simply that w depends on the value of cf. $\tau_{w_{\text{induction}}}$ and $\tau_{w_{\text{recovery}}}$ are time constants. We assume that the induced change is maintained for much longer time than the time necessary for the induction, so that $\tau_{w_{\text{induction}}} \ll \tau_{w_{\text{recovery}}}$.

The value of v is initialized to $v(0)$ and updated by a Hebbian rule with a postsynaptic gating mechanism [13] as follows:

$$\tau_v \dot{v} = \mathrm{mf} \cdot (\mathrm{vn} - \theta) \tag{7}$$

$$\tau_\theta \dot{\theta} = -\theta + \mathrm{vn}, \tag{8}$$

where θ calculates the running average of vn. Owing to this parameter, the plasticity rule updates the value of v towards either potentiation or depression, depending on the history of the activity of vn. Thus, θ provides a gating mechanism. τ_θ and τ_v are time constants and assumed to be $\tau_\theta \ll \tau_v$.

Inputs to this network are given by mf and cf. We assume that mf represents the average activity of MFs, and that the value in resting state is almost the same with that during training. Eventually, we regard mf as constant. We discuss on the soundness of this assumption in Discussion. Other variables are functions of time t.

3 Results

Computer Simulation of VOR Gain Adaptation

We carried out computer simulations of VOR gain adaptation. Initially, we performed a freerun for 10,000 steps to set parameters to their equilibrium states (data not shown). Then, we carried out a course of 4 sessions of training, where the first 2 sessions are for gain up training and the others for gain down training. Each session is composed of a training period for 100 steps and a post-training period for 1,900 steps. Parameters were set as follows: mf = 1, cf = −1 for gain up training and +1 for gain down training, $w(0) = 1$, $v(0) = 1$, $\tau_{w_{\text{induction}}} = 10$, $\tau_{w_{\text{recovery}}} = 500$, $\tau_\theta = 500$, and $\tau_v = 2,000$.

Figure 2 plots the change of the values of w, v and vn throughout the 4 sessions. The value of w repeats twice to decrease quickly during training and recover slowly after the training, and then repeats twice to increase quickly during training and recover slowly after the training, suggesting that w is short-term memory. The value of v gradually increases during the first 2 sessions and then gradually decreases during the latter 2 sessions, as if v accumulates the change of w throughout the 4 sessions. In other words, the value of w is transferred to as that of v during post-training periods, suggesting memory transfer from the cortex to the nuclei and consolidation of v as long-term memory. The value of vn increases quickly during training and decays slowly after the training. Owing to the memory transfer, the baseline value of vn changes throughout the 4 sessions.

Figure 3 plots the value of θ and vn across the 4 sessions. For each session, the value of vn increases quickly and then decreases slowly. The value of θ increases gradually and exceeds that of vn at some point during post-training periods. Thereafter, $(\text{vn} - \theta)$ in Eq. (7) becomes negative and converges to zero in time. Therefore, the value of v does not diverge to infinity. This result suggests that the postsynaptic factor θ acts as a constraint for robust long-term memory formation to prevent v from diverging.

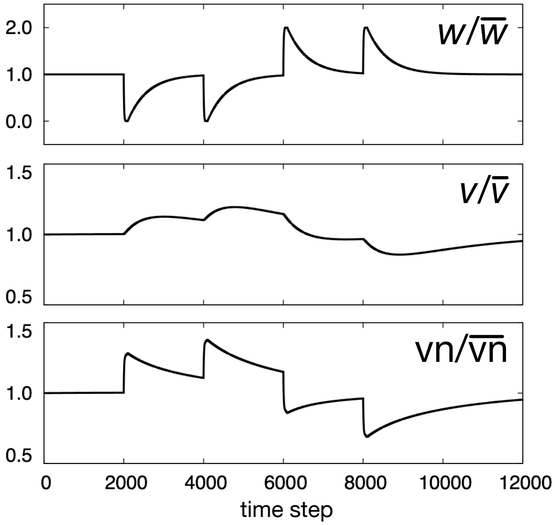


Fig. 2. Plots of w (top), v (middle) and vn (bottom) in a course of 4 sessions of training. They are divided by $\bar{w}$, $\bar{v}$ and $\bar{\text{vn}}$, the values at their equilibrium state, respectively, and normalized.

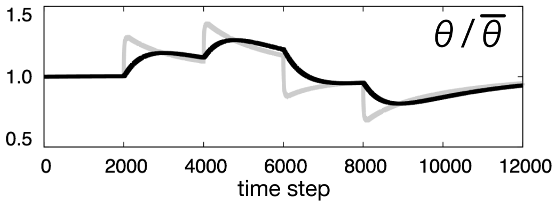


Fig. 3. Plots of θ (black) and vn (gray) in the course of 4 sessions of training. Both θ and vn are normalized as in Fig. 2.

Disruption of Memory Transfer by Blocking Purkinje Cells' Activity During Post-training Periods

In the cerebellar cortex, molecular interneurons such as stellate and basket cells exert GABAergic inhibition to Purkinje cells. Injection of a GABA agonist muscimol into the cerebellar cortex may enhance GABA receptors on Purkinje cells, and thereby blocking Purkinje cells' activity. Pharmacological studies have shown that muscimol injection into the cerebellar cortex after training disrupts memory transfer [9, 11], suggesting that spontaneous activity of Purkinje cells after training is necessary for memory transfer.

We carried out simulation of this post-training blockade of Purkinje cells' activity. For each post-training period, we replaced Eq. (4) with

$$\mathrm{vn} = v \cdot \mathrm{mf}, \tag{9}$$

by omitting the inhibition term.

Figure. 4 shows the result. By blocking Purkinje cells' activity after each training, the value of vn immediately goes to the almost initial level, because the short-term memory stored in the cerebellar cortex is shut down. On the other hand, because the value of θ is still high, the term $(\mathrm{vn} - \theta)$ in Eq. (7) becomes large negative, resulting in the decrease of v to the almost initial level. Therefore, the value of vn does not exhibit accumulative change across the 4 sessions, indicating that long-term memory formation is disrupted.

4 Discussion

We proposed a model on post-training memory transfer in cerebellar motor learning. In our model, synaptic weights of MF-nuclear cell synapses are updated by a Hebbian rule with a postsynaptic gating mechanism, which is controlled by the running average of the postsynaptic neuron activity. We demonstrated that memory transfer occurs mainly after training, not during training, and spontaneous activity of Purkinje cells after training is necessary for memory transfer. These results are consistent with experimental results [9–11].

Previous theoretical studies have demonstrated that the conventional Hebbian rule adopted for MF-nuclear cell synapses, in which the synaptic weight is

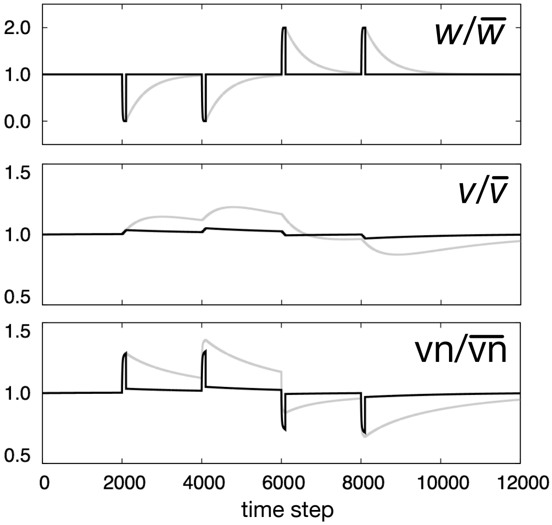


Fig. 4. Plots of w (top), v (middle) and vn (bottom) in a course of 4 sessions of training under the blockade of Purkinje cells' activity during post-training periods. Gray lines are identical to those in Fig. 2 for comparison. Conventions as in Fig. 2.

updated by the correlated activity of MFs and the nuclei, allows the weight value to diverge to infinity, indicating the failure of memory transfer [14, 15]. Instead, these studies have proposed Purkinje cell-dependent rule, in which the synaptic weight is updated by the correlated activity of MFs and Purkinje cells innervating to the nuclei. Purkinje cell-dependent rule is demonstrated to prevent the synaptic weight from divergence. A problem, however, is that the biological mechanism to update the synaptic weight between a pre- and a post-synaptic neurons with the help of a third neuron is unclear. On the other hand, our model based on a Hebbian rule does not need a third neuron, which allows to update the synaptic weight using local information only. This is an advantage of our model over the previous studies. Another problem of the previous studies is that their models show memory transfer during training, not after training. This is another advantage of the present model.

In addition to the Hebbian rule with a postsynaptic gating mechanism employed for MF-nuclear cell synapses, two more assumptions are made in our model. First assumption is that the average activity of MFs is almost the same with that during training. Semicircular canal neurons that provide MF inputs in VOR elicit spikes about 60 spikes/s spontaneously and increase or decrease their firing rates equally depending on the direction of head rotation [16]. This observation implies that the average firing rate is always equal to the spontaneous firing rate, which could justify our assumption. Second assumption is that induced plastic change at PF-Purkinje cell synapses is maintained for much longer time than the time necessary for the induction. In slice experiments, LTD at PF-Purkinje cell synapses is induced by paired stimulation of PFs at 4 Hz and CFs at 1 Hz for 5 minutes. The LTD persists more than 1 hour [12]. This observation could justify the second assumption as well. To maintain induced LTD, protein synthesis on Purkinje cells is necessary [17]. Application of anisomycin, a protein synthesis inhibitor, to the cerebellar cortex after training disrupts memory transfer [10]. Our model is consistent with this experimental result.

In sum, our model provides a clue from a theoretical viewpoint to settle the controversy on the location of motor memory in the cerebellum [1].

References

1. Melvill, J.G.: Motor learning in vestibulo-ocular control. In: Kandel, E.R., Schwartz, J.H., Jessell, T.M. (eds.) Principles of Neural Science, pp. 824–828. McGraw-Hill (2000)
2. DufosséIto, M., Jastreboff, P.J.M., Miyashita, Y.: A neuronal correlate in rabbit's cerebellum to adaptive modification of the vestibulo-ocular reflex. Brain Res. 150, 611–616 (1978)
3. Miles, F.A.: Lisberger SG Plasticity in vestibulo-ocular reflex: a new hypothesis. Ann. Rev. Neurosci. 4, 273–299 (1981)
4. Ito, M.: Cerebellar control of the vestibulo-ocular reflex-Around the flocculus hypothesis. Ann. Rev. Neurosci. 5, 275–297 (1982)
5. Nagao, S., Kitazawa, H.: Effects of reversible shutdown of the monkey flocculus on the retention of adaptation of the horizontal vestibulo-ocular reflex. Neuroscience 118, 563–570 (2003)
6. Kassardjian, C.D., Tan, Y.F., Chung, J.Y., Heskin, R., Peterson, M.J., Broussard, D.M.: The site of a motor memory shifts with consolidation. J. Neurosci. 25, 7979–7985 (2005)
7. Shutoh, F., Ohki, M., Kitazawa, H., Itohara, S., Nagao, S.: Memory trace of motor learning shifts transsynaptically from cerebellar cortex to nuclei for consolidation. Neurosci. 139, 767–777 (2006)
8. Anzai, M., Kitazawa, H., Nagao, S.: Effects of reversible pharmacological shutdown of cerebellar flocculus on the memory of long-term horizontal vestibulo-ocular reflex adaptation in monkeys. Neurosci. Res. 68, 191–198 (2010)
9. Attwell, P.J.E., Cook, S.F., Yeo, C.H.: Cerebellar Function in Consolidation of a Motor Memory. Neuron 34, 1011–1020 (2002)
10. Okamoto, T., Endo, S., Shirao, T., Nagao, S.: Role of cerebellar cortical protein synthesis in transfer of memory trace of cerebellum-dependent motor learning. J. Neurosci. 31, 8958–8966 (2011)
11. Okamoto, T., Shirao, T., Shutoh, F., Suzuki, T., Nagao, S.: Post-training cerebellar cortical activity plays an important role for consolidation of memory of cerebellum-dependent motor learning. Neurosci. Lett. 504, 53–56 (2011)
12. Ito, M.: Long-term depression. Ann. Rev. Neurosci. 12, 85–102 (1989)
13. Gerstner, W., Kistler, W.: Spiking Neuron Models. Cambridge University Press (2002)
14. Medina, J.F., Mauk, M.D.: Simulations of cerebellar motor learning: Computational analysis of plasticity at the mossy fiber to deep cerebellar nucleus. J. Neurosci. 19, 7140–7151 (1999)

15. Masuda, N., Amari, S.: A computational study of synaptic mechanisms of partial memory transfer in cerebellar vestibulo-ocular-reflex learning. J. Comput. Neurosci. 24, 137–156 (2008)
16. Ezure, K., Schor, R.H., Yoshida, K.: The response of horizontal semicircular canal afferents to sinusoidal rotation in the cat. Exp. Brain Res. 33, 27–39 (1978)
17. Linden, D.J.: A protein synthesis-dependent late phase of cerebellar long-term depression. Neuron. 17, 483–490 (1996)

Surface-Based Construction of Curvature Selectivity from the Integration of Local Orientations

Yasuhiro Hatori[1,2] and Ko Sakai[1]

[1] Graduate School of Information Engineering, University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan
hatori@cvs.cs.tsukuba.ac.jp, sakai@cs.tsukuba.ac.jp
[2] Research Fellow of the Japan Society for the Promotion of Science

Abstract. Recent physiological studies have reported that neurons in the cortical area V4 are selective to curvature along object contour. The neurons are capable of discriminating convexity and concavity, and indicating the direction of the curvature with respect to the contour projected onto their cRF. We propose that surface representation plays a crucial role in constructing the selectivity for curvature because convexity/concavity cannot be determined without the construction of object surface. To test the proposal, we developed a computational-model of V1-V4 networks that computes spatiotemporal activities of single cells, and carried out the simulations with the stimuli used by the physiological studies. The model neurons reproduced the selectivity for specific curvature and direction. Population of the model cells showed a bias toward convex curvature as consistent with V4 population *in vivo*. These results support that the representation of surface is crucial for the construction of the selectivity in V4.

Keywords: shape representation, surface, curvature selectivity.

1 Introduction

Accumulated evidence of physiology has suggested that representation of object shape is gradually constructed through multiple visual cortices in the ventral stream [1-4]. In the early-stage of the stream, a retinal image is decomposed by Gabor filters into local features such as oriented line-segment, and these features are gradually integrated in order by intermediate-stages (V2-V4) of the stream. Recent physiological studies have suggested that neurons in V4 bind oriented line-segments and construct the representation of curvature along object contour [3, 4]. A precise analysis of the dynamics of the curvature-selective neurons has revealed that the driving feature (of stimulus) gradually changes from local orientation to a combination of the orientations [5]. Specifically, early-phase responses correspond to orientation fragment and late-phase responses correspond to a combination of the orientations. It is of great interest what neural mechanisms bind the local features to construct a higher representation of shape.

We propose a hypothesis that higher shape representation is constructed from the integration of local features based on the surface represented by V2. Specifically, the

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 425–433, 2012.

present study addresses how the selectivity for curvature in V4 emerges from orientations represented in V1. There are two advantages to utilize surface for the integration: (i) it preserves the convexity/concavity even if other cues (e.g., local contrast & color) are identical (see Fig. 1a), and (ii) temporal dynamics (early- and late-phases) of V4 neurons agrees with the latency of orientation extraction in V1 and that of surface representation in V2 [6, 7].

To test our proposal, we developed a computational model of V1-V2-V4 networks, with all possible combinations of local orientations (in V1) and its locations (for details, see the next section). This thorough combination enabled us to exclude any explicit mechanism for reproducing physiological results. We carried out the simulations of the model with the stimuli defined by curvature and its direction, the same set as that used in the physiological study [3]. The results showed that the model single cells reproduced the selectivity for curvature and its direction, and that population activity was biased toward acute curvature, showing a good agreement with physiology. These results support that the representation of surface in early visual areas plays a crucial role for the construction of the curvature selectivity in V4.

2 The Model

The activity of a model V4 neuron is determined by two distinct terms of early- and late-phases: (i) summation of V1 cells' activity (early-phase), (ii) binding the V1 activity with a specific combination of orientations with respect to the surface (late-phase). Fig. 2 shows a schematic illustration of the spatial correspondence between

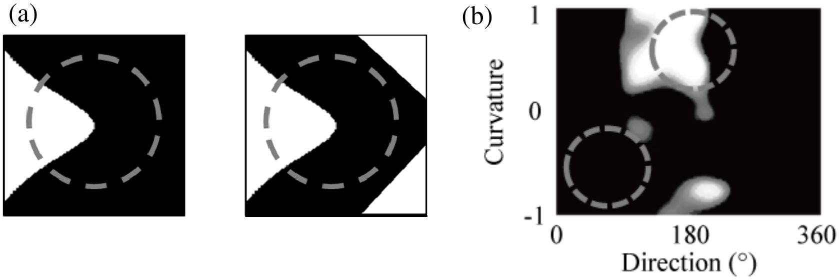


Fig. 1. The role of surface in producing the selectivity for curvature. (a) Convexity/concavity of a contour cannot be determined without the representation of surface. Psychophysics shows that observers often see a white region as a figure (surface) in the left panel while black region in the right panel. Such assignment of surface is necessary to signal convexity/concavity, because local structures (indicated by dashed circles) are identical between the two panels. (b) An example of the response distribution of a single V4 neuron as a function of curvature and its direction with respect to the cRF [replotted from 4]. This cell responded strongly to convex curvature (positive in ordinate) at the left of the cRF (180°). Therefore, this cell will respond to a stimulus similar to the left panel in (a)(the corresponding range is denoted by a dashed circle at the top of (b)) but not to the right panel (denoted by a dashed circle at the bottom of (b)).

stimulus and the classical receptive fields (cRFs) of the model cells (Fig. 2a), and an overview of the computation in V1 and V4 (Fig. 2b). Note that this model lacks explicit implementation of V2 where the representation of surface emerges [6, 7]. Because, we aim to focus on the construction of the curvature selectivity, but not to study the construction of surface. Although there is no explicit V2 layer in the model, we assume that V2 is included in the process of constructing the surface representation.

2.1 V1 Layer

This layer realizes the orientation selectivity in V1 cells by the convolution with 16 oriented Gabor filters followed by half-wave rectification and divisive normalization [8]. Contrast (C_θ) at each spatial position (*x,y*) is given by:

$$C_\theta(x,y) = (G_\theta * I)\,(x,y), \tag{1}$$

where * indicates convolution, *I* is input, G_θ is Gabor filter orientated in θ degrees. The contrast passes through half-wave rectification and divisive normalization to evoke the output of this stage (O_θ^{V1}). Divisive normalization [8] is given by:

$$O_\theta^{V1}(x,y) = {C_\theta(x,y)}\Big/\big(a + wT(x,y)\big)' \tag{2}$$

$$T(x,y) = \textstyle\sum_\theta C_\theta(x,y), \tag{3}$$

where *a* and *w* are constants for preventing division by zero [8]. We set *a* = *w* = 1 for the present simulations.

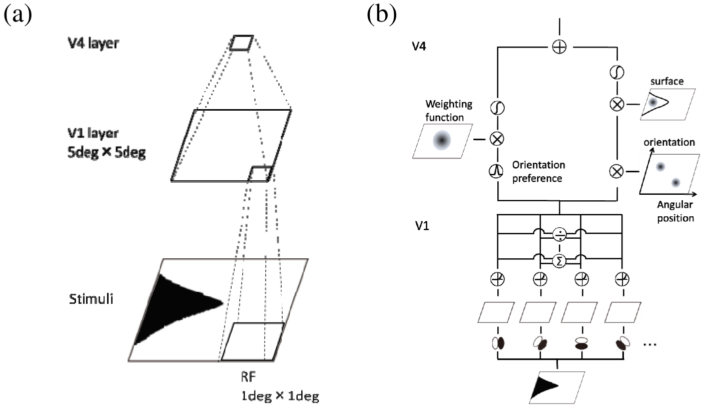


Fig. 2. An illustration of the model V1-V4 network. (a) Spatial correspondence among a stimulus and the RFs of cells in V1 and V4. (b) An overview of computation. See details for the text.

2.2 V4 Layer

The activity of a single V4 cell is computed in two phases. The early-phase response is a spatial summation of V1 cells' activity and the late-phase response is determined by a combination of orientations at specific positions with respect to object surface. It is expected that tuning for curvature is obtained by the late-phase response.

Early-Phase Response: During the early-phase, the model computes a linear summation of the activities of specific V1 cells. We defined the response during the early phase as:

$$T_{Early}^{V4} = \sum_{pref \in \Theta} \left(Gauss * O_{pref}^{V1}\right)(x, y), \tag{4}$$

$$\Theta = \{0°, 22.5°, \cdots, 337.5°\}, \tag{5}$$

where *Gauss* represents a Gaussian defined by the center position of RF (x_0, y_0; equivalent to the center of a stimulus) and standard deviation (SD; σ = 20pixels, approximately 1 degree in visual angle). Θ is a set of 16 orientations. *pref* represents preferred two orientations out of 16 orientations, which is inherent in each cell. The early-phase response passes through a sigmoidal function to realize compressive nonlinearity observed in early vision. We described the output of this phase (O_{Early}^{V4}) as:

$$O_{Early}^{V4} = sig\left(T_{Early}^{V4}\right), \tag{6}$$

$$sig(T) = {}^{asym}\!\big/_{\left(1 + e^{-(T - shift)gain}\right)}, \tag{7}$$

where *shift*, *gain* and *asym* are constants that determine origin, slope and asymptote of the sigmoid, respectively. We set these constants empirically as *shift* = 0.35, *gain* = 10, and *asym* = 1 so as to realize the compressive nonlinearity and limit the output within a range (from 0 to 1).

Late-Phase Response: The tuning for curvature is obtained by the late-phase where activities of V1 cells are bound (Fig. 3). The bind is based on the direction of surface and the combination of orientations (both of which are given as preference of a model V4 cell). For example, a cell whose preference is denoted in Fig. 3a shows a strong response to a protrusion (defined by the orientations) projected onto the right (defined by the surface direction; Fig. 3c). We define the late-phase response as:

$$T_{Late}^{V4} = \sum_{\theta \in \Theta} sig(w \times A_\theta), \tag{8}$$

where *w* indicates the weight that represents the preference for the position of surface center with respect to the cRF center. We set constants of sigmoidal function as *shift* = 0.125, *gain* = 100, and *asym* = 0.5. A_θ represents the summation of O_θ^{V1} weighted by the preference for angular positions and orientations ($Comb_{ORI,ANG}(\theta, \varphi)$) described as:

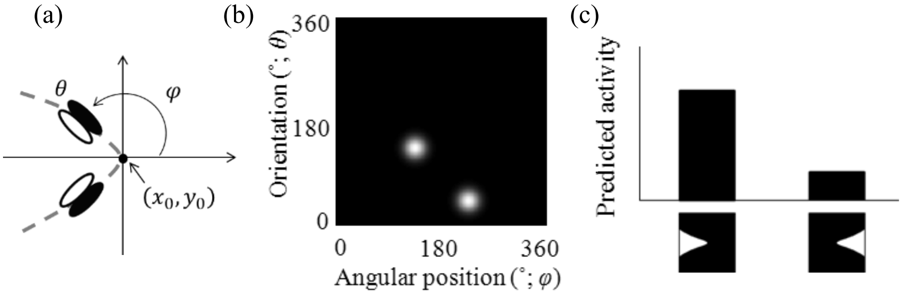


Fig. 3. Predicted late-phase response of the model. (a) Example preference for a combination of orientations. The center of the cRF is located at the origin. (b) The preferred combination denoted in (a) is shown as a function of orientation and its angular position. (c) The predicted activities of the model cell denoted in (a).

$$A_\theta = Comb_{ORI,ANG}(\theta, \varphi) \sum_{x,y} O_\theta^{V1}(x, y), \tag{9}$$

$$Comb_{ORI,ANG}(\theta, \varphi) = \sum_{i=1}^{N} {}^{exp\left(-\frac{(\theta - ori_i) + (\varphi - ang_i)}{2\sigma^2}\right)}\Big/_{2\pi\sigma}\ , \tag{10}$$

where *ORI* and *ANG* represent preferred combinations of orientation and its angular position with respect to the cRF center, respectively. ori_i and ang_i indicate *i*th preference of a cell for the orientation and its angular position. We used *N* = 2 and *σ* = 5°. An example of *Comb* is shown in Fig. 3b. There are two distinct tunings that are defined by the angular position and the orientation (eq.10). All possible combinations of two orientations (from 16 orientations) and two angular positions (from 8 positions) were used in the simulations, leading to 9,216 (16×16×8×9/2; same combination of angular positions are excluded from the simulations) distinct combinations (model cells). φ is the angular position of O_θ^{V1} with respect to the center of cRF and is calculated by:

$$\varphi = \ arctan\left({}^{\left(M_y^\theta - y_0\right)}\Big/_{\left(M_x^\theta - x_0\right)} \right), \tag{11}$$

where $\left(M_x^\theta, M_y^\theta\right)$ represent the position of centroid of O_θ^{V1}. (x_0, y_0) represent the center position of cRF (equivalent to the center of a stimulus). We define the late-phase response (O_{Late}^{V4}) as:

$$O_{Late}^{V4} = sig(T_{Late}^{V4}), \tag{12}$$

where we set *shift = 0.7*, *gain* = 10, and *asym* = 1.

Response of Single V4 Neurons: The activity of a model V4 cell (O^{V4}) is given by the summation of the early-phase response and the late-phase response as:

$$O^{V4} = O_{Early}^{V4} + O_{Late}^{V4}. \tag{13}$$

3 Results

To examine whether the representation of surface is crucial for the construction of the curvature selectivity from local orientations, we carried out the simulations of the model with the stimuli (Fig. 4) defined by curvature (squashed within 0-1) and its direction (0°-357.5°). Stimulus size was 108×108 pixels, and we defined 21 pixels as 1 degree in visual angle. These stimuli were comparable to those used in physiological studies so that we can compare the results of simulations and physiological experiments in single-cell and population behavior.

3.1 Single Cell Behavior

We compared the tuning for curvature and direction between the model single-cells and V4 neurons *in vivo* reported by Carlson et al [4]. Fig. 5 shows the activity of model single-cells (Fig. 5a-c) and that of V4 neuron (Fig. 5d; [4]). A model single-cell (Fig. 5a) shows the strongest response to the stimulus of which curvature is 1 (sharp convex) and its direction is 70°, yielding a clear tuning for acute curvature on the upper right of the object. Other model neurons (Fig.5b & 5c) show a distinct tuning for curvature and its direction with similar tuning widths. For example, a model cell in Fig. 5c is selective to obtuse curvature. The tuning of single V4 neurons reported by Carlson et al. showed (1) the bias for acute curvature, and (2) the anisotropy in the direction of curvature (Fig. 5d; [4]). The model cells showed the similar characteristics, as observed in Fig. 5a to 5c. These results indicate that the model reproduces the behavior of V4 neurons at single-cell level.

3.2 Population Behavior

We demonstrated that behavior of model single cells agrees with that of real V4 neurons. Here, we examined whether population behavior agrees with physiological data. Specifically, Carlson et al. have reported that a population activity of V4 neurons (obtained from a summation of single neurons' activity) showed the bias toward acute curvature (cells respond more frequently to the stimuli of which curvature is above 0.3) with isotropy in its direction (no bias in the direction of curvature), as shown in Fig. 6b ([4]; n = 165). A summed activity of all model cells is shown in Fig. 6a (n = 9,216). The population activity of model cells shows a bias toward acute curvature, with the isotropy in its direction, as similar to the physiological data. Note that we prepared the model with all possible combinations of the preferences in local orientations and positions. It is natural to expect no bias in the preference. Therefore, our results are surprising and provide a strong constrain on the understanding of the underlying neural mechanisms. Our further analysis suggests that the activity of a majority of model cells were rather weak and their preference was biased toward acute curvature, while the model cells with stronger responses do not. We expect that further analysis will lead to the further understanding of population behavior.

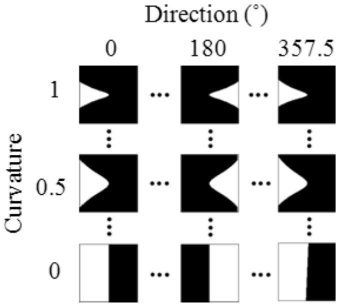


Fig. 4. Stimuli used in the simulations. These are defined by curvature (ranges from 0 to 1) and its direction (0° - 357.5°; 0° indicates protrusion toward right).

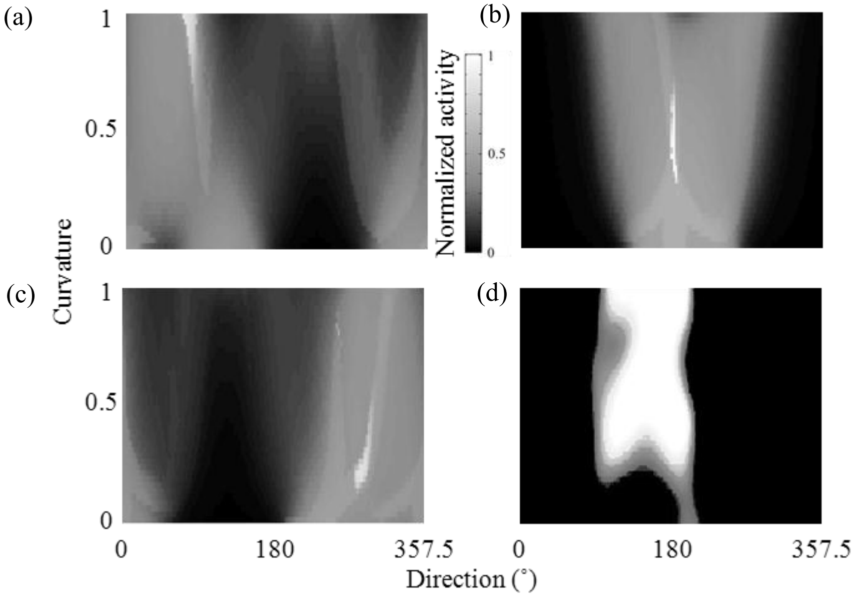


Fig. 5. Example responses of model cells (a-c) and a V4 neuron *in vivo* (d) reported by Carlson et al [4] are plotted along the curvature (vertical axis) and its direction (horizontal axis). Response to stimulus is denoted by grey (whiter represents stronger activity).

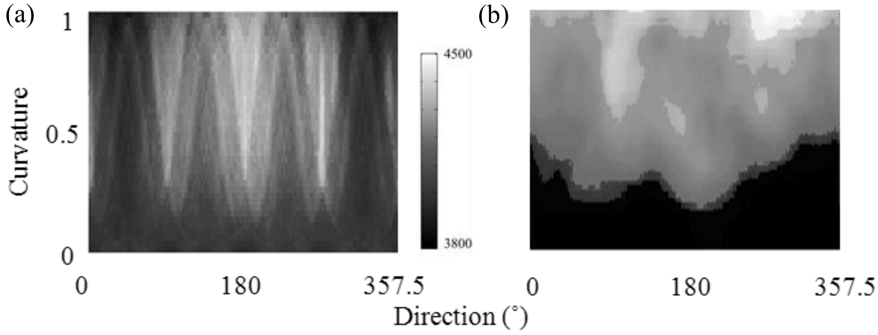


Fig. 6. Population activity of model cells (a) and V4 neurons reported by Carlson et al [4] (b). The conventions are the same as Fig. 5. The activities are obtained from the summation of single neurons' activity.

4 Conclusions and Discussions

We proposed that the curvature selectivity is constructed by combining local orientations based on surface representation. To test the proposal, we developed a computational model that computes the activities of single V4 cells, and carried out the simulations of the model with the stimuli defined by curvature and its direction. Simulation results revealed that the model single-cells reproduced the selectivity for curvature and its direction, and that population activity was biased toward acute curvature, indicating good agreements with physiological data [3, 4].

Lack of the explicit implementation of V2 layer does not affect the validity of the model. Although the V2 layer potentially influences the activities of the early- and late-phase, we considered only the early-phase in the present model. We assumed that V2 layer is included in the process of emergence of surface representation during the late-phase. If V2 layer were explicitly implemented in the computation of the early-phase, the activity of a V2 cell was given by a spatial pooling of V1 cells' activities. In the present model, a similar pooling is performed in V1-V4 connections with a pooling area larger than V1-V2. The larger pooling in V1-V4 should be equivalent to a cascade of smaller pooling in V1-V2 and V2-V4. Although BO-selective cells in V2 might have non-uniform pooling, a V4 cell would pool all of them to yield unconditional pooling. Therefore, the implementation of V2 layer have no (or a small) effect to the computation of early-phase activity unless pooling includes strong nonlinearity.

Our model accounts for crucial, physiological characteristics of V4 neurons. First, the model differentiates its activities to convexity from concavity as same as real neurons [4]. Cadieu et al. have proposed a computational model for shape selectivity in V4 [9]. Their model was based on a weighted sum of local orientations with various spatial positions, spatial frequencies and sizes. Although their model can predict the selectivity of V4 neurons, the model did not capture the crucial characteristic of the neurons: their model cannot distinguish convex from concave. Our model is capable of distinguishing convexity and concavity because surface defines the direction of figure so that either convexity or concavity is uniquely determined. Second, time course of the model V4 neuron is consistent with a physiological study of Yau et al [5]. They demonstrated that the activity of V4 cells was characterized by two distinct phases. The two phases of the model (early- and late-phase) agree with those of the actual neurons. The distinct phases are not reproduced by a weighted summation of orientations without surface representation [9], spectral receptive field model [10], and filter-rectify-filter pathway model [11], because these models lack multiple computational pathways of which time courses are different. Note that our model is not designed to reproduce the tuning of V4 cells for color [12] and disparity [13] so that other mechanisms are needed for explaining these characteristics. These results support that the representation of surface in early visual areas is crucial for the construction of curvature selectivity that leads to the representation of shape.

Acknowledgement. This work was supported by grant-in aids from JSPS and MEXT of Japan (KAKENHI 22300090, 24.3368, and 23135503 (Shitsukan)).

References

1. Hubel, D.H., Wiesel, T.N.: Receptive Fields and Functional Architecture of Monkey Striate Cortex. J. Physiol. 195, 215–243 (1968)
2. Ito, M., Komatsu, H.: Representation of Angles Embedded within Contour Stimuli in Area V2 of Macaque Monkeys. J. Neurosci. 24, 3313–3324 (2004)
3. Pasupathy, A., Connor, C.E.: Responses to Contour Features in Macaque Area V4. J. Neurophysiol. 82, 2490–2502 (1999)
4. Carlson, E.T., Rasquinha, R.J., Zhang, K., Connor, C.E.: A Sparse Object Coding Scheme in Area V4. Curr. Biol. 21, 288–293 (2011)
5. Yau, J.M., Pasupathy, A., Brincat, S.L., Connor, C.E.: Curvature Processing Dynamics in Macaque Area V4. Cereb. Cortex (2012)
6. Zipser, K., Lamme, V.A.F., Schiller, P.H.: Contextual Modulation in Primary Visual Cortex. J. Neurosci. 16, 7376–7389 (1996)
7. Bakin, J.S., Nakayama, K., Gilbert, C.D.: Visual Responses in Monkey Areas V1 and V2 to Three-Dimensional Surface Configurations. J. Neurosci. 20, 8188–8198 (2000)
8. Carandini, M., Heeger, D.J.: Normalization as a Canonical Neural Computation. Nat. Rev. Neurosci. 13, 51–62 (2011)
9. Cadieu, C., Kouh, M., Pasupathy, A., Connor, C.E., Risenhuber, M., Poggio, T.: A Model of V4 Shape Selectivity and Invariance. J. Neurophysiol. 98, 1733–1750 (2007)
10. David, S.V., Hayden, B.Y., Gallant, J.L.: Spectral Receptive Field Properties Explain Shape Selectivity in Area V4
11. Wilson, H.R., Wilkinson, F.: Detection of global structure in Glass patterns: implications for form vision. Vis. Res. 38, 2933–2947 (1998)
12. Schein, S.J., Desimone, R.: Spectral Properties of V4 Neurons in the Macaque. J. Neurosci. 10, 3369–3389 (1990)
13. Hinkle, D.A., Connor, C.E.: Quantitative Characterization of Disparity Tuning in Ventral Pathway Area V4. J. Neurophysiol. 94, 2726–2737 (2005)

Solving Dynamic Constraint Optimization Problems Using ICHEA

Anurag Sharma and Dharmendra Sharma

Faculty of Information Sciences and Engineering
University of Canberra, ACT, Australia
{Anurag.Sharma,Dharmendra.Sharma}@canberra.edu.au

Abstract. Many real-world constrained problems have a set of predefined static constraints that can be solved by evolutionary algorithms (EAs) whereas some problems have dynamic constraints that may change over time or may be received by the problem solver at run time. Recently there has been some interest in academic research for solving continuous dynamic constraint optimization problems (DCOPs) where some new benchmark problems have been proposed. Intelligent constraint handling evolutionary algorithm (ICHEA) is demonstrated to be a versatile constraints guided EA for continuous constrained problems which efficiently solves constraint satisfaction problems (CSPs) in [22], constraint optimization problems (COPs) in [23] and dynamic constraint satisfaction problems (DCSPs) in [24]. We investigate efficiency of ICHEA in solving benchmark DCOPs and compare and contrast its performance with other well-known EAs.

Keywords: evolutionary algorithm (EA), constraint satisfaction problem (CSP), dynamic constraint satisfaction problems (DCSP), constraint optimization problem (COP), dynamic constraint optimization problem (DCOP), Intelligent constraint handling evolutionary algorithm (ICHEA).

1 Introduction

Many engineering problems ranging from resource allocation and scheduling to fault diagnosis and design involve constraints that must be satisfied to have an acceptable solution. Some of these constraints and/or objective function can change over time which makes the problem more complex like ship scheduling, vehicle routing, dynamic obstacle avoidance, the adaptive farming strategies and aerodynamic/structural wing design problems [5, 16, 21]. The constraint problems can be divided into four classes: static constraint satisfaction problems (CSPs), dynamic constraint satisfaction problems (DCSPs), static constraint optimization problems (COPs) and dynamic constraint optimization problems (DCOPs). The difference between static/dynamic constraint optimization and constraint satisfaction is that in first an optimal solution that satisfies all the constraints available at that particular time should be found, while in second any solution as long as all the constraints

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 434–444, 2012.

available at the given time are satisfied is acceptable [7]. This paper concentrates only on DCOPs.

EAs have been successful in solving many static COPs where objective function of non-contained optimization problem is generally bundled with problem dependent penalty functions. A penalty term is used in general for reward and punishment for satisfying and/or violating the constraints [4] where the aim is to decrease (*punish*) the fitness of infeasible solutions as to favor those feasible individuals in the selection and replacement processes. The main advantage of the use of penalty functions is their simplicity; however, their main shortcoming is that penalty factors, which determine the severity of the punishment, must be set by the user and their values are problem dependent that requires a careful fine-tuning of parameter to obtain competitive results [12, 13]. There are some other novel approaches in the literature to handle static constraints effectively. Some of the important relevant approaches applied in constraint handling for EAs are summarized below from [4, 10].

Some other constraint handling approaches include expensive *repair* algorithms that promote the local search to transform infeasible solutions to feasible solutions because the feasible parents not necessarily produce feasible progenies [4]. In multi-objective optimization (MOO) constraints are transformed into multiple objectives. There are many established MOO algorithms like MOGA [8], VEGA [20], NSGA and NSGAII [6]. Paredis in [17] has used co-evolution strategies that utilizes *predator-prey* model to keep two populations – one population represents solutions that satisfies many constraints while other population represents those individuals whose constraint(s) is violated by lots of individuals in the first population. This strategy requires extra computational effort to find the intersection of a line with the boundary of the feasible region.

The use of domain knowledge within an EA can also be utilized to improve its performance as EAs are 'blind' to constraints. Recently, there have been few algorithms developed that move away from penalty based fitness functions to generic distance function given in Eq. (8). ICHEA [22] uses its *intermarriage* crossover operator to look for overlapping feasible regions through differentiating the boundaries of feasible regions for each constraint. This reduces the search space to obtain the solution efficiently. Cultural algorithms are also used to extract domain knowledge for its evolutionary search by using two subpopulations – population space and the belief space. Ricardo and Carlos in [18] proposed cultured differential evolution (CDE) that uses differential evolution (DE) as the population space and belief space as the information repository to store experiences of individuals for other individuals to learn. Amirjanov in [1] proposed changing domain range based genetic algorithm (CRGA) that adaptively shifts and shrinks the size of search space of the feasible region by employing feasible and infeasible solution in the population to reach the global optimum. Mezura-Montes et. al. in [14] proposed simple multi-membered evolution strategy (SMES) that uses a simple diversity mechanism by allowing infeasible solutions to remain in the population. A simple feasibility-based comparison mechanism is used to guide the process toward the feasible region of the search space. The idea is to allow the individual with the lowest amount of constraint violation and the best value of the objective function to be selected for the next population. PSO-DE proposed by [12] is another algorithm that integrates particle swarm optimization (PSO) and DE to solve real-valued COPs.

A DCOP is a sequence of static constraints added, removed or updated in the search space of the problem. It is indeed easy to see that all the possible changes (constraint or domain modifications, variable additions or removals) can be expressed in terms of constraint additions or removals [26]. To solve such a sequence of constraints, it is always possible to solve each one from scratch as it has been done for the first one but this naive method, which remembers nothing from the previous reasoning, has two important drawbacks [26]:

- **Inefficiency:** which may be unacceptable in the framework of real time applications (planning, scheduling etc), where the time allowed for re-planning is limited.
- **Instability:** of the successive solutions, which may be unpleasant in the framework of an interactive design or a planning activity, if some work has been started on the basis of the previous solution

A major question raises here is whether all the constraint handling approaches for static COPs are applicable for DCOPs as well [16]. This question has not been addressed extensively in the literature especially for real-valued DCOPs where benchmark problems were also unavailable unit Nguyen and Yao has introduced some problems in [15, 16] together with a penalty function based novel algorithm repair genetic algorithm (RepairGA) to solve these problems efficiently. Richter in [19] proposed memory design to solve DCOPs. Memory design is traditionally used to solve unconstrained dynamic optimization problems where the usual practice is to set aside a memory space to hold some promising individuals from the population that replaces other poor performing individuals when the change in environment is detected [19]. There are also two canonical EAs namely hyper-mutation genetic algorithm (hyperM) and random-immigrant genetic algorithm (RIGA) that are based on "introduce diversity" and "maintain diversity" strategies respectively to solve DCOPs. We have used the same benchmark problems to test the performance of ICHEA against RepairGA, hyperM, RIGA and canonical genetic algorithm (GA). There are some benchmark problems for dynamic optimization problems in [11] and [9] that are without constraints where some recently developed EAs have performed well on these benchmark problems like self-adaptive differential evolution algorithm (*jDE*) [3], dynamic hybrid particle swarm optimization (*DHPSO*) [9] and triggered memory based PSO (*TMPSO*) [27].

This paper is organized as follows: Section 2 describes the mathematical formalization of real valued DCOPs. Section 3 briefly revisits ICHEA from [22–24] with addition to its applicability in handling DCOPs. Section 4 shows experimental results of ICHEA with other state-of-the-art EAs to solve benchmark DCOPs with analysis about the results in Section 5. Section 6 concludes the paper by summarizing the results and proposing some further extensions to the research.

2 Formalization of Real-Valued DCOPs

A solution to real-valued static COP has two folds – search for an optimum solution that also must satisfy all the constraints. Real-valued COP can be formulated as:

$$\text{optimize } f(\vec{x}) \tag{1}$$

where COP's objective function $f(\vec{x})$ has an n-dimensional input vector $\vec{x} = \{x_1, x_2, \dots x_n\}$ that is defined in a search space S. More specifically, $\vec{x} \in \mathcal{F} \subseteq S$, where $\mathcal{F}$ being the feasible region on the search space $S \subseteq \mathbb{R}^n$. Usually, the search space S is defined as a n-dimensional rectangle in $\mathbb{R}^n$. The domain of variables are defined by their lower bounds l_i and upper bounds u_i:

$$l_i \leq x_i \leq u_i, \quad 1 \leq i \leq n \tag{2}$$

The feasible region $\mathcal{F}$ with bounds on each dimension is further restricted by a set of m additional constraints that can be given in two relational forms – equality and inequality [6, 12, 25].

$$g_i(\vec{x}) \geq 0 \qquad i = 1, \dots, k \tag{3}$$

$$h_j(\vec{x}) = 0 \quad j = k + 1, \dots, m \tag{4}$$

The equality constraints $h_j(\vec{x})$ cannot be solved directly using EAs so it is converted into inequality constraints by introducing a positive tolerance value δ.

$$g_j(\vec{x}) = \delta - |h_j(\vec{x})| \geq 0 \tag{5}$$

A set of individual feasible regions $\{\mathcal{F}_1, \mathcal{F}_2, ..\mathcal{F}_m\}$ for each constraint can also be defined as:

$$\mathcal{F}_i = \{\vec{x} \in \mathcal{F} \mid g_i(\vec{x}) \geq 0, 1 \leq i \leq m, i \in Z\} \tag{6}$$

where Z is the set of integers. Many EAs uses a distance function as their fitness function to rank individuals. The distance function indicates how far a chromosome is from the feasible regions [6]. This fitness function tries to bring the chromosomes closer to the feasible region using the following function for $\forall i : \{1 \leq i \leq m\}$:

$$fitness_i(\vec{x}) = \begin{cases} g_i(\vec{x}), & if\ g_i(\vec{x}) < 0 \\ 0, & if\ g_i(\vec{x}) \geq 0 \end{cases} \tag{7}$$

$$e = \sum_{i=1}^{m} |fitness_i(\vec{x})| \tag{8}$$

The fitness function fitness_i is a measurement of *euclidean* distance of a vector $\vec{\text{x}}$ from a feasible region $\mathcal{F}_\text{i}$. The error function e is the summation of all the fitness functions. Minimizing the error value e leads toward a CSP solution where the objective function $f(\vec{x})$ is not needed. A solution to CSP is found when $e = 0$. To get a COP solution, CSP solutions are further processed to get optimum value of $\vec{\text{x}}$ that optimizes the objective function $f(\vec{x})$. For DCOPs the total number of constraints m is not know a priori and the solution has to be produced based on constraints that come to hand where the constraints and objective functions can change over time. Hence the fitness functions for static COP given in Eq. (7) and Eq. (8) has to be transformed into dynamic COP by making it time dependent by introducing a parameter for time *t*.

$$fitness_i(\vec{x}, t) = \begin{cases} g_i(\overrightarrow{x}, t), & if\ g_i(\vec{x}, t) < 0 \\ 0, & if\ g_i(\vec{x}, t) \geq 0 \end{cases} \tag{9}$$

$$e(t) = \sum_{i=1}^{m(t)} |fitness_i(\vec{x}, t)| \tag{10}$$

where $g_i(\vec{x}, t)$ is inequality constraint function at time t that delivers fitness $fitness_i(\vec{x}, t)$. Its dynamic CSP solution with total number of constraints $m(t)$ at time t is given by $e(t)$.

To determine the performance of an algorithm at time t is usually measured by an offline performance measure described in [2, 27] which is an average fitness error between the optimal fitness of the current environment and the best-of-generation fitness at each generation. The average fitness error $(error(t))$ at time t can be calculated as:

$$error(t) = \frac{1}{t} \sum_{i=1}^{t} \left| f_{optimal} - f_{best}(t) \right| \tag{11}$$

where $f_{optimal}$ is the known optimal fitness and $f_{best}(t)$ is the fitness of the best solution achieved at generation t. Nguyen and Yao in [15] has modified this offline performance that demonstrates the performance based on "good feasible solution" rather than any good solution that may be infeasible. This measure is always greater than or equal to zero. If in any generation there is no feasible solution, the worst possible value that a feasible solution can have is taken for $f_{best}(t)$; however, this can be incomputable for some hard problems (for example problems discussed in [22]) where any feasible solution may not be found for the entire generations. Fortunately, all of these benchmark problems in [15] are able to find at least one feasible solution easily before the change in environment. We used both of these offline errors for our experiments and to differentiate these measurements the first one is called *traditional performance measure* and later one is called *feasibility performance measure*. Feasibility performance measure is more applicable for constrained problems as it takes feasibility into account because infeasible solutions are not acceptable for constrained problems.

3 ICHEA for DCOPs

ICHEA is a variation of EA that is an effective and versatile constraint handling tool that has been demonstrated to outperform other EAs to solve hard benchmark CSPs in [22] and DCSPs in [24]. It has also shown very competitive results for benchmark COPs in [23]. ICHEA uses its novel search operator *intermarriage* crossover that uses knowledge from constraints rather than blindly searching for the solution. In this crossover both parents belong to different feasible regions $\mathcal{F}_i$ and $\mathcal{F}_j$ where $i \neq j$. It is also possible that a parent does not belong to any of the feasible regions $S - F$. These parents are made to come closer towards the boundary of their corresponding feasible regions to locate the overlapping regions where more constraints are satisfied. Please refer to [22–24] for details about this operator. The generated offspring from *intermarriage* crossover contains genes from both parents. The purpose is to make a "generic" offspring that tries to satisfy more than one constraint because its parents are from two different feasible regions. The algorithm favours those offspring which satisfy more constraints by utilizing Deb's ranking scheme based on feasibility [6] to rank the population where the population is first sorted according to number of satisfied constraints in decreasing order then by fitness value given in Eq. (10) in increasing order. The pseudocode of ICHEA can be given as:

```
chromosomes  = initializeChromosomes();
for each generation
  parents = NoveltyTournamentSelection();
  offspring = interMarriageCrossover(parents);
  Mutation(offspring);
  chromosomes = chromosomes + offspring;
  SortAndReplace();
  ResolveLocalOptimalSolutions();
  CheckTerminationCriteria();
End for loop;
```

The description of algorithm and subroutines can be found in [22–24].

4 Experiments

As mentioned in Section 1, Nguyen and Yao have proposed some new benchmark problems in [16] which we will be using to compare the performance of ICHEA with aforementioned dynamic constraint handling algorithms RepairGA, RIGA, hyperM and GA. The description of test problems is given in Table 1 where some problems have been omitted because of their common properties with the listed ones and unavailability of published results. Please note that the problem G24_3 has a typographical error for variable $S_2(t)$ which should be $S_2(t) = 2 - t\frac{x_2max - x_2min}{S}$. The paramter settings for RepairGA, RIGA, hyperM and GA have been kept same as in the published results in [16]. ICHEA is a problem independent tool and it does not require any penalty function. The parameters settings to solve all the benchark problems are: Population = 100, crossover rate = 1.0, muation rate = 0.1, $|P_{dimMatrix}| = 2$.

An average of 10 successive runs for ICHEA is taken into account to demonstrate its solution quality against published results of above mentioned algorithms in [16]. The problem environment changes after every 1000 evaluations as in [16] which is approximately 40 generations. To compare the performance of ICHEA with other algorithms we used both performance measures given in Eq. (11). The experimental

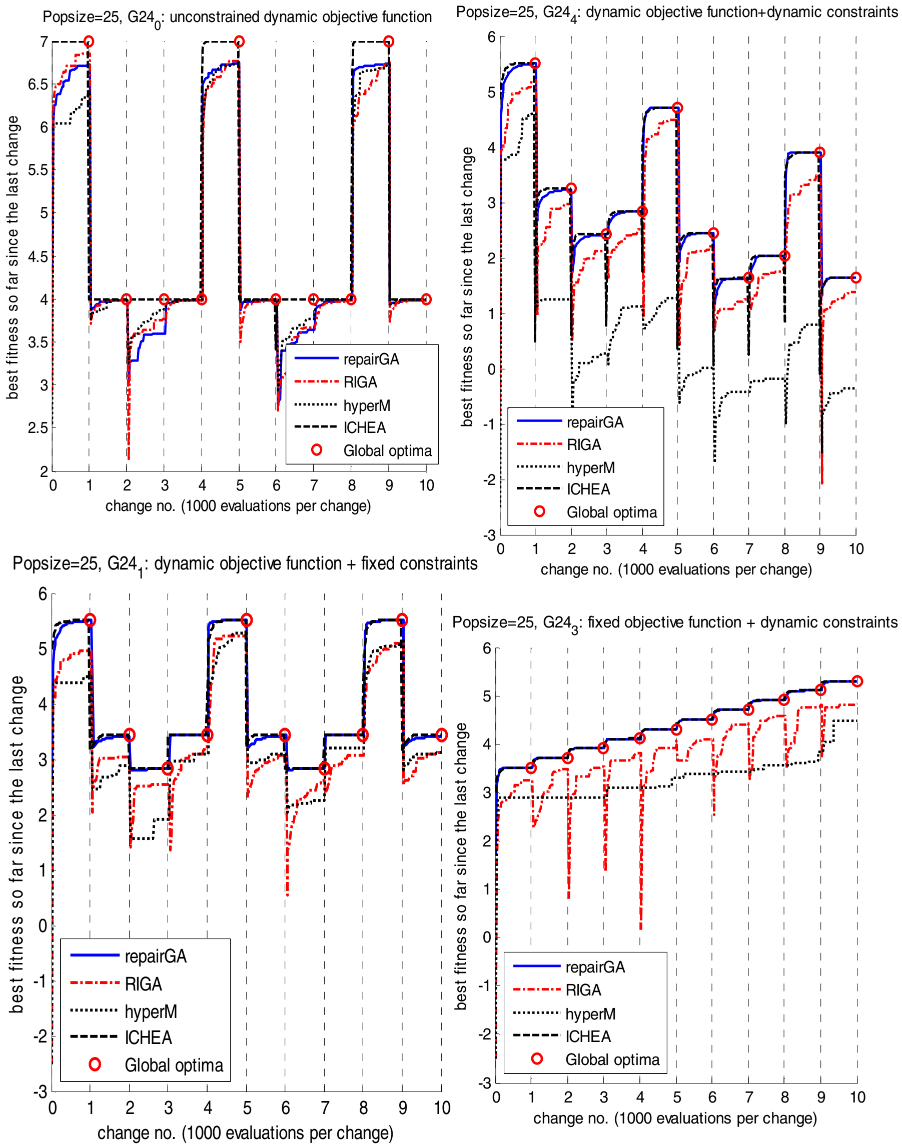


Fig. 1. Plots of current best solution on each generation for repairGA, RIGA, hyperM and ICHEA

Table 1. Properties of Benchmark Problems [21]

Problem	objFunc	Constr	DFR	SwGO	GIB	NAO
G24_0	Cyclic	No	-	-	Yes	No
G24_1	Cyclic	Fixed	2	Yes	Yes	No
G24_3	Fixed	Linear	1-3	Yes	Yes	Yes
G24_4	Cyclic	Linear	1-3	Yes	Yes	No
DFR	Number of disconnected feasible regions					
SwGO	Global opt. switches among disconnected regions					
NAO	Newly appearing optima without changing existing optima					
GIB	Global optimum is in the boundary of feasible area					

Table 2. Traditional Performance Measure

Algorithms	Error	StDev	vsGA	Error	StDev	vsGA
	G24_0 (dynF + noC)			G24_1 (dynF + fixC)		
ICHEA	0.0051	0.004	88.44	0.0333	0.005	24.35
RepairGA	0.2531	0.026	1.77	0.0448	0.009	18.13
RIGA	0.2854	0.043	1.57	0.5734	0.076	1.42
HyperM	0.2660	0.012	1.69	0.6472	0.271	1.25
GA	0.4488	0.049	-	0.8117	0.077	-
	G24_3 (fixF + dynC)			G24_4 (dynF+dynC)		
ICHEA	0.0187	0.003	52.24	0.0799	0.006	11.07
RepairGA	0.0148	0.002	66.11	0.0695	0.009	12.72
RIGA	0.6664	0.063	1.46	0.5937	0.054	1.49
HyperM	1.1079	0.482	0.88	2.3370	1.942	0.38
GA	0.9760	0.127	-	0.8842	0.081	-

Table 3. Feasibility Performance Measure

Algorithms	Error	StDev	Error	StDev
	G24_0 (dynF + noC)		G24_1 (dynF + fixC)	
ICHEA	0.005	0.004	0.033	0.005
RepairGA	0.468	0.059	0.226	0.024
RIGA	0.131	0.034	0.401	0.046
HyperM	0.173	0.042	0.450	0.094
GA	0.214	0.037	0.587	0.085
	G24_3 (fixF + dynC)		G24_4 (dynF+dynC)	
ICHEA	0.019	0.003	0.066	0.009
RepairGA	0.116	0.008	0.211	0.015
RIGA	0.340	0.045	0.492	0.071
HyperM	0.461	0.104	0.494	0.039
GA	0.384	0.092	0.627	0.045

result of the traditional performance measure is given in Table 2 and feasibility performance measure is given in Table 3. Both tables show the error value as their performance value with standard deviation (StDev) and comparison measurement with GA (vsGA) that indicates how many times the tested algorithms is better than GA. Table 2 shows ICHEA has outperformed other algorithms on test problem G24_0 and G24_1, and showed similar performance with RepairGA for G24_3 and G24_4 based on traditional performance measurements while Table 3 shows the feasibility performance measure which is more applicable measurement in the context of constrained problems as infeasible solutions are not taken into account where ICHEA has completely outperformed all other algorithms on all the tested problems. Plots of the current best individuals for every generation for all the algorithms are shown in Fig. 1.

5 Discussion

All the tested benchmark problems are unique in nature in the context of DCOPs. G24_0 objective function changes over time that does not have any constraint, G24_1 also has dynamic objective function but fixed constraints. The constraints of G24_3 and G24_4 change over time but G24_3's objective function is fixed while G24_4's objective function is dynamic. The performance comparison based on traditional performance measure shows ICHEA has outperformed all other algorithms where problems have no constraints or constraints are fixed. For dynamic constraints ICHEA has produced similar performance as of RepairGA; however, this traditional measurement does not reflect the quality of solutions in terms of their feasibility. Hence feasibility performance measure is more applicable when there are constraints in the problem. The test results based on feasibility performance measure shows that ICHEA has clearly outperformed all other algorithms because ICHEA's main strength is that it makes use of knowledge from constraints rather than blindly search in the solution space as traditional EAs do [22–24]. Fig. 1 and Table 3 show ICHEA is not only able to immediately recover from change in environment but also produces good quality feasible solutions. ICHEA runs two parallel processes internally – one to keep looking for feasible solutions and another to optimize the existing feasible solutions [23]. Any disruption in the search space invokes ICHEA to prioritize search for feasible solutions through its *intermarriage* crossover that results in high quality feasible solutions. Another advantage of ICHEA is that it does not use problem dependent or problem class dependent penalty functions as RepairGA uses penalty value of 2.5 for these benchmark problems.

6 Conclusion

ICHEA is a versatile EA to solve various types of real-valued constraint problems. It has already been demonstrated to perform well for benchmark CSPs, DCSPs and COPs [22–24]. The aim of this paper is to evaluate the performance of ICHEA on newly proposed benchmark DCOPs where it has outperformed other state-of-the-art EAs on the scale of feasibility performance measurement; however, there is need to diversify the benchmark problems with high dimensional problems as current problems are limited to two dimensional only. An advantage of ICHEA over other EAs is that it is a problem independent constraint handling EA that utilizes knowledge from constraints without using any repair or penalty functions. It extracts and exploits knowledge from constraints for its evolutionary search and is independent of the characteristics of the problems. For future work ICHEA still needs to be tested on discrete constraint problems as it has all the potential to perform well. The enhancement will also include *incrementailty* in search through changing constraints and addition/retraction of constraints.

Acknowledgment. We would like to thank Dr. Trung Thanh Nguen of The Centre of Excellence for Research in Computation Intelligence and Applications (CERCIA), School of Computer Science, University of Birmingham, B15 2TT, United Kingdom for providing the data files from his experimental results from [16].

References

1. Amirjanov, A.: The development of a changing range genetic algorithm. Computer Methods in Applied Mechanics and Engineering 195, 2495–2508 (2006)
2. Branke, J.: Designing evolutionary algorithms for dynamic optimization problems. In: Advances in Evolutionary Computing: Theory and Applications, pp. 239–262 (2003)
3. Brest, J., et al.: Dynamic optimization using Self-Adaptive Differential Evolution. In: IEEE Congress on Evolutionary Computation, CEC 2009, pp. 415–422 (2009)
4. Coello Coello, C.A.: Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer Methods in Applied Mechanics and Engineering 191, 1245–1287 (2002)
5. Craenen, B.G.W., et al.: Comparing evolutionary algorithms on binary constraint satisfaction problems. IEEE Transactions on Evolutionary Computation 7, 424–444 (2003)
6. Deb, K., et al.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
7. Eiben, A.E.: Evolutionary Algorithms and Constraint Satisfaction: Definitions, Survey, Methodology, and Research Directions. In: Theoretical Aspects of Evolutionary Computing, pp. 13–58 (2001)
8. Fonseca, C.M., Fleming, P.J.: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Proceedings of the 5th International Conference on Genetic Algorithms, pp. 416–423 (1993)
9. Karimi, J., et al.: A new hybrid approach for dynamic continuous optimization problems. Applied Soft Computing 12, 1158–1167 (2012)
10. Kramer, O., Schwefel, H.-P.: On three new approaches to handle constraints within evolution strategies. Nat. Comput. 5, 363–385 (2006)
11. Li, C., et al.: Benchmark Generator for CEC'2009 Competition on Dynamic Optimization (2008)
12. Liu, H., et al.: Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl. Soft Comput., 629–640 (2010)
13. Mezura-montes, E., Coello Coello, C.A.: A Survey of Constraint-Handling Techniques Based on Evolutionary Multiobjective Optimization. Departamento de Computación, Evolutionary Computation Group at CINVESTAV (2006)
14. Mezura-Montes, E., Coello Coello, C.A.: A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Transactions on Evolutionary Computation 9, 1–17 (2005)
15. Nguyen, T.T., Yao, X.: Continuous Dynamic Constrained Optimisation - The Challenges. IEEE Transactions on Evolutionary Computation 99 (2012)
16. Nguyen, T.T., Yao, X.: Benchmarking and solving dynamic constrained problems. In: IEEE Congress on Evolutionary Computation, CEC 2009, pp. 690–697 (2009)

17. Paredis, J.: Co-evolutionary Constraint Satisfaction. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, Springer, Heidelberg (1994)
18. Becerra, R.L., Coello Coello, C.A.: Cultured differential evolution for constrained optimization. Computer Methods in Applied Mechanics and Engineering 195, 4303–4322 (2006)
19. Richter, H.: Memory Design for Constrained Dynamic Optimization Problems. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcazar, A.I., Goh, C.-K., Merelo, J.J., Neri, F., Preuß, M., Togelius, J., Yannakakis, G.N. (eds.) EvoApplicatons 2010, Part I. LNCS, vol. 6024, pp. 552–561. Springer, Heidelberg (2010)
20. Schaffer, J.D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In: Proceedings of the 1st International Conference on Genetic Algorithms, pp. 93–100 (1985)
21. Shang, Y., Fromherz, M.P.J.: Experimental complexity analysis of continuous constraint satisfaction problems. Information Sciences 153, 1–36 (2003)
22. Sharma, A., Sharma, D.: ICHEA – A Constraint Guided Search for Improving Evolutionary Algorithms. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part I. LNCS, vol. 7663, pp. 269–279. Springer, Heidelberg (2012)
23. Sharma, A., Sharma, D.: Real-Valued Constraint Optimization with ICHEA. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part III. LNCS, vol. 7665, pp. 406–416. Springer, Heidelberg (2012)
24. Sharma, A., Sharma, D.: An Incremental Approach to Solving Dynamic Constraint Satisfaction Problems. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part III. LNCS, vol. 7665, pp. 445–455. Springer, Heidelberg (2012)
25. Tessema, B., Yen, G.G.: A Self Adaptive Penalty Function Based Algorithm for Constrained Optimization. In: IEEE Congress on Evolutionary Computation, CEC 2006, pp. 246–253 (2006)
26. Verfaillie, G., Jussien, N.: Constraint Solving in Uncertain and Dynamic Environments: A Survey. Constraints, 253–281 (2005)
27. Wang, H., Wang, D.-W., Yang, S.: Triggered Memory-Based Swarm Optimization in Dynamic Environments. In: Giacobini, M. (ed.) EvoWorkshops 2007. LNCS, vol. 4448, pp. 637–646. Springer, Heidelberg (2007)

An Incremental Approach to Solving Dynamic Constraint Satisfaction Problems

Anurag Sharma and Dharmendra Sharma

Faculty of Information Sciences and Engineering
University of Canberra, ACT, Australia
{Anurag.Sharma,Dharmendra.Sharma}@canberra.edu.au

Abstract. Constraint satisfaction problems (CSPs) underpin many science and engineering applications. Recently introduced *intelligent constraint handling evolutionary algorithm* (ICHEA) in [14] has demonstrated strong potential in solving them through evolutionary algorithms (EAs). ICHEA outperforms many other evolutionary algorithms to solve CSPs with respect to success rate (SR) and efficiency. This paper is an enhancement of ICHEA to improve its efficiency and SR further by an enhancement of the algorithm to deal with local optima obstacles. The enhancement also includes a capability to handle dynamically introduced constraints without restarting the whole algorithm that uses the knowledge from already solved constraints using an incremental approach. Experiments on benchmark CSPs adapted as dynamic CSPs has shown very promising results.

Keywords: Constraint satisfaction problem (CSP), intelligent constraint handling evolutionary algorithm (ICHEA), evolutionary algorithm (EA), local optima, dynamic constraints, incremental approach.

1 Introduction

CSPs are at the core of many real-world applications, including control, and diagnosis of physical systems such as car, planes, and factories. It is also used in modern robotic systems such as control of modular, hyper-redundant robots, which are robots with many more degrees of freedom than required for typical tasks. Sometimes the environment of the CSPs changes along with time as in obstacle avoidance, vehicle routing and reusing previously generated university timetable. Even though CSPs are an important area of research in part of computer science, little has been reported on the development of efficient and effective constraint-handling techniques – relative to the development of new methods for unconstrained optimization using EAs [7]. Recently introduced ICHEA [14] is able to solve real-valued CSPs efficiently with relatively higher success rate (SR) than other tested well known EAs. The strength of ICHEA is that it makes use of knowledge from constraints rather than blindly search in the solution space as done by traditional EAs [2]. ICHEA has been demonstrated to outperform other well regarded EAs like NSGA II [3] and PSO-DE [10]. However it also exhibited drawbacks in solving hard CSPs where it was computationally

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 445–455, 2012.

expensive as its median solutions to benchmark CSPs generally required more than 200.0 seconds of CPU time to solve on a common standalone machine [14]. Its SR was also very low in the range of 0.0-0.6 for hard problems. Hence we have introduced some new strategies to improve ICHEA to address these drawbacks.

Moreover, ICHEA is only able to handle static CSPs in its current state so we propose an enhancement to the ICHEA to realize *incrementality* in constraint solving. Using this approach the dynamic behavior of CSPs can be handled efficiently as it is quintessential for real-time dynamic CSPs (DCSP) where only little research has been reported using EAs. Furthermore, there are not any established benchmark problems for them. Some benchmark problems are compiled in [12] and [6] but they are used for dynamic COPs and dynamic optimization problems respectively. The difference between COPs and CSPs is that in first an optimal solution that satisfies all the constraints should be found, while in second any solution as long as all the constraints are satisfied is acceptable [4]. Because of the unavailability of benchmark DCSPs we have transformed some existing benchmarks CSPs from [15] to DCSPs.

The main contribution of this paper is to enhance the performance of existing ICHEA (called ICHEA+) to solve CSPs and introduce an incremental approach to solve real-valued DCSPs using ICHEA. The paper is organized as follows: Section 2 describes the formalization of CSPs and DCSPs. Section 3 briefly discusses EA techniques used to handle dynamic behavior of CSPs (called I-ICHEA) and the available benchmark problems. Section 4 describes the enhancement of ICHEA with some new strategies to overcome getting local optimal solutions. Section 5 shows experimental results with discussions in Section 6 about the outcome. Section 7 concludes the paper by summarizing the results and proposing some further possible extensions to the research.

2 Formalization of CSPs and DCSPs

A CSP is defined by an n dimensional input vector $X = \{x_1, x_2, \dots\ x_n\}$ in a finite space S where each variable x_i has a finite domain D_i. A set of m constraints $\{c_1, c_2, \dots\ c_m\}$ are defined in the form of functions:

$$c_i(x_1, x_2, \dots\ x_n) = \begin{cases} 1, & if\ satisfied \\ 0, & if\ violated \end{cases} \tag{1}$$

Constraint satisfaction sets $\{S_1, S_2, .. S_m\}$ can also be defined where:

$$S_i = \{X \in S \mid c_i(X) = 1, 1 \le i \le m, i \in Z\} \tag{2}$$

where Z is the set of integers. The solution of a CSP is $s \in S$ when all the constraints c_i are satisfied, which can be given as:

$$\textstyle\sum_{i=1}^{m} c_i(s) = m \tag{3}$$

For real-valued CSPs numerical constraints can be given in two forms – equality and inequality functions [3, 10, 16]:

$$g_i(X) \ge 0 \qquad i = 1, \dots, k \tag{4}$$

$$h_j(X) = 0 \qquad j = k + 1, \dots, m \tag{5}$$

The equality constraints cannot be solved directly using EAs so they are converted to inequality constraints by introducing a positive tolerance value δ.

$$g_j(X) = \delta - \left|h_j(X)\right| \geq 0 \tag{6}$$

Generally violation count is used as a fitness function for any CSPs. Depending on the strengths of constraints, individual weights is assigned to constraints in a *penalty* function to calculate the fitness value. To avoid using problem dependent *penalty* functions and utilizing some knowledge from constraints to guide the evolutionary search many EAs do not use violation count but use a distance function to indicate how far an individual is from the feasible regions [11]. It transforms the inequality constraint functions to a fitness function to rank individual members in the population generated by ICHEA. This fitness function tries to bring the individuals closer to the feasible space using the following functions for $\forall i : \{1 \leq i \leq m\}$:

$$fitness_i(X) = \begin{cases} g_i(X), & if\ g_i(X) < 0 \\ 0, & if\ g_i(X) \geq 0 \end{cases} \tag{7}$$

$$e = \sum_{i=1}^{m} \left|fitness_i(X)\right| \tag{8}$$

The fitness function $fitness_i$ is a measurement of *euclidean* distance of vector X from the nearest point of the feasible region where constraint c_i is satisfied. The error function e is the summation of all the fitness functions. The objective is to minimize the error value e where the solution to a CSP is found when $e = 0$.

For DCSPs the total number of constraints m is not know a priori and the solution has to be produced based on constraints that come to hand. A DCSP can be defined as a sequence of static CSPs where each one differs from the previous one by the addition or removal of some constraints. It is indeed easy to see that all the possible changes to a CSP (constraint or domain modifications, variable additions or removals) can be expressed in terms of constraint additions or removals [17]. The same fitness function given in Eq. (7) and Eq. (8) are used for DCSPs. To solve such a sequence of CSPs, it is always possible to solve each constraint from scratch as it has been done for the first one but this naive method, which remembers nothing from the previous reasoning, has two significant drawbacks [17]:

— **Inefficiency:** which may be unacceptable in the framework of real time applications (planning, scheduling etc), where the time allowed for re-planning is limited.

— **Instability:** of the successive solutions, which may be unpleasant in the framework of an interactive design or a planning activity, if some work has been started on the basis of the previous solution.

A DCSP is a sequence of static CSPs that are formed by constraint changes. The notion of DCSP has been introduced to represent such situations by [13]. Some attempt has been made to solve DCSP using EA as [5] uses Multi-objective optimization (MOO) to transform changes in constraints as a new objective function with changes in so called *disruption* function. This function is used to estimate the effect of changing an initial constraint to a new one. The changes are reflected in *pareto set* and program runs again to get the new *pareto* optimal set guided by previous *pareto* front. In a typical MOO problem there exists a set of solutions which are superior to the rest of the solution in the search space when all objectives are considered but are inferior to other solutions in the space in one or more objectives. A local search is another approach to reuse previous solutions for DCSP. The previous solution can simply be used as a starting assignment for the new local search repair-based algorithm. DCSP features employing the previous related CSP to find a minimal change solution to the current CSP but it can be computationally challenging [9, 17].

3 Benchmark Problems

As mentioned above there is little research reported on real-valued DCSPs nor there is any benchmark problems available for it. There are some benchmark problems for dynamic optimization problems in [8] and [6] that are without constraints. Some recently developed EAs have performed well on these benchmark problems like self-adaptive differential evolution algorithm (jDE) [1], dynamic hybrid particle swarm optimization (DHPSO) [6] and triggered memory based PSO (TMPSO) [18]. Nguen and Yao in [12] has introduced some benchmark problems for real-valued dynamic COPs and a novel algorithm repair genetic algorithm (RepairGA) to solve these problems efficiently; however, none of benchmarks are for DCSPs. We took some benchmark CSPs from coconut benchmark [15] and converted them to DCSP by taking one constraint at a time that is solved as a sequence of static constraints. In this paper only addition of constraints are considered to make a dynamic environment. Update of constraints or redefinition of feasible regions has not been considered. A new constraint is added into the environment in every 100 generations or else if all the current constraints are satisfied – whichever is earlier.

4 Enhancement to ICHEA

ICHEA is a variation of EA that uses its own crossover operator namely *intermarriage* crossover that selects two parents from different *constraint satisfaction sets* S_i to make them come closer iteratively towards their corresponding feasible boundary because the CSP solutions lie in the overlapping boundary region of feasible regions that satisfy different constraints. Favoring individuals that satisfy higher number of constraints and the use of feasible regions for *intermarriage* crossover guide the evolutionary search in finding the solution space quickly [14]. This guiding process has helped ICHEA to outperform other well-known EAs to solve CSPs where constraint strengths are very high i.e. the feasible regions are very small compared to the whole search space. Calculation for constraint strengths has been shown in Section 5.

As mentioned in Section 1, even after ICHEA's success in outperforming other well-known EAs to solve CSP, it is still computationally expensive as its median solutions for some benchmark problems generally require more than 200.0 seconds of a CPU time on a common machine to produce a solution [14]. Its SR is also very low in the range of 0.0-0.6 for some hard benchmark problems. Hence we propose the following enhancement that improves its performance in terms of efficiency and SR. The enhanced ICHEA is called ICHEA+ (ICHEA-plus) where the improvement observed has been as high as 68 times over the previous ICHEA on benchmark problems. ICHEA+ is even able to produce efficient solutions with high SR of up to 1.00 on low positive tolerance value ($\delta = 10^{-3}$) on hard CSP problems where previous ICHEA had low success with SR = 0.00.

4.1 Diversity Management

According to [10] the lower the individuals' degree of constraint violation, the higher the probability that it clusters together around the current best solution and individuals with lower degrees of constraint violations are very difficult to jump out of current best individual's adjacent region. This may cause the current best individual to stay on the same position for a long time leading to loss of diversity in the population. To avoid this scenario the ICHEA+ keeps the fair share of all degrees of constraint violating individuals in the population. If the population pop of size $|pop|$ has m constraints in the problem then the whole population is divided into equal sized m slots where slot i is allocated to individuals that violate i constraints. If there are no individuals with i violations then its allocated space is evenly distributed to other slots. Let pop_i indicate the population of individuals that violate i constraints so the total population is:

$$pop = \sum_{i=1}^{m} pop_i$$

Then pop_i is sorted according to the fitness and the best $|pop|/m$ is selected for subpopulation pop_i.

$$\therefore \max(|pop_i|) = |pop|/m$$

If after allocation, k slots have $|pop_i| < |pop|/m$, then unallocated population of individuals $pop_{unalloc}$ is:

$$pop_{unalloc} = \sum_{i=1}^{m} \begin{cases} |pop|/m - |pop_i|, & if\ |pop_i| < |pop|/m \\ 0 & , \ otherwise \end{cases}$$

This unallocated population $pop_{unalloc}$ needs to be allocated evenly in the slots that have $|pop_i| > |pop|/m$.

4.2 Stalled Local Optimal Solutions Management

The above diversity management is not sufficient to avoid the population getting stuck into local optimal solution for hard CSPs. This is a common problem for EAs when the whole population gets stuck around local optimal solution and lose its diversity. We introduce the concept of *forced constraint violations* to tackle this issue. This works like *tabu* search algorithm where the individuals try to move away from the forcibly introduced new infeasible regions (*tabu* regions). If the population is stagnant for certain number of generations then the current best solution is considered as local optimal solution where some region around it is marked as a new infeasible region to move the population away from it. This region is defined as a hyper-sphere whose centre is the location of the current best (local optimal) solution with the radius defined as distance from the location of current best individual with the location of the worst individual that has the same degree of violations as the current best individual. If the current best individual belongs to a subpopulation pop_i which is sorted according to the fitness from best to worst where an individual can be described as $X_j \in \{pop_i | 1 \leq j \leq |pop_i|\}$ has best individual X_1 and worst individual $X_{|pop_i|}$. The radius R of the hyper-sphere can be computed as: $R = |X_1 - X_{|pop_i|}|$ and hence the new forced dynamic constraint is: $g_{m+1}(X) = \sum_{i=1}^{n}(x_i - \mu_i)^2 > R^2$ where $\mu_i \in X_1$ and $x_i \in$ X. Fig. 1 demonstrates the movement of the current best individual that starts from high violation regions towards low violation regions until it is trapped in a stagnant region which is then referred as stalled local optimal solution.

5 Experiments

ICHEA is a problem independent tool to solve any given n dimensional CSP so we use the following parameters to solve all the problems:
Stall threshold = 12 generations, crossover rate = 1.0, mutation rate = 0.1, maximum generation = 1000 and $|pop| = \begin{cases} 25, & \text{n} > 6 \\ 100, & 1 \leq \text{n} \leq 6 \end{cases}$

Table 1. Benchmark Quadratic Problem Chem

δ		ICHEA+	I-ICHEA	ICHEA	imp
10^{-1}	SR	1.00	1.00	1.00	0.0
	Best	11 gens at 1.8s	19 gens at 2.7s	54 gens at 0.83s	0.5
	Median	26 gens at 4.03s	25 gens at 3.5s	238 gens at 4.66s	1.2
	Worst	37 gens at 6.7s	33 gens at 4.8s	559 gens at 11.1s	1.7
10^{-3}	SR	1.00	1.00	0.30	0.7
	Best	75 gens at 7.3s	34 gens at 5.4s	5900 gens at 196.4s	26.9
	Median	621 gens at 108.3s	267 gens at 45.7s	-	3.1
	Worst	740 gens at 122.9s	291 gens at 49.6s	-	2.7

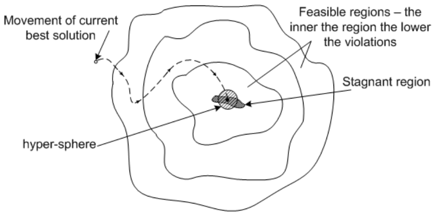

Movement of current best solution
Feasible regions – the inner the region the lower the violations
Stagnant region
hyper-sphere

Fig. 1. Making hyper-sphere around stalled local optimal solution

Table 2. COP Benchmark problem G05

δ		ICHEA+	I-CHEA	ICHEA	imp
10^{-5}	SR	1.00	1.00	1.00	0.0
	Best	26 gens at 0.52s	29 gens at 0.53s	18 gens 0.40s	0.77
	Median	30 gens at 0.57s	34 gens at 0.62s	19 gens 0.41s	0.72
	Worst	39 gens at 0.72s	36 gens at 0.64s	21 gens at 0.46s	0.64

As one constraint is considered at a time for DCSP, we would also like to see if the constraint strengths of individual constraint matters in finding an efficient solution. Hence two different sequences of static CSPs are used where each constraint is added into the environment – from lowest to highest strength and vice versa. As described in [10] constraint strength (ρ) are computed *offline* by using the formula $= |\Omega|/|pop|$, where $|pop|$ is the number of solutions randomly generated from pop,

Table 3. Benchmark Trigonometric Problem HS109

δ		ICHEA+	I-ICHEA ($\rho\uparrow$)	I-ICHEA ($\rho\downarrow$)	ICHEA	imp
10^{-1}	SR	0.70	0.70	0.70	0.70	0.0
	Best	54 gens at 81.1s	57 gens at 87.7s	59 gens at 79.8s	53 gens at 71.0s	0.9
	Median	131 gens at 205.0s	113 gens at 183.0s	66 gens at 92.1s	70 gens at 239s	1.2
	Worst	192 gens at 361.3s	186 gens at 283.6s	150 gens at 208.6s	-	2.8
10^{-3}	SR	0.10	0.80	0.80	0.0	0.8
	Best	133 gens at 204.7s	100 gens at 155.0s	89 gens at 128.4s	-	4.9
	Median	-	122 gens at 186.0s	125 gens at 183.2s	-	0.0
	Worst	-	156 gens at 250.5s	151 gens at 230.6s	-	0.0

Table 4. Benchmark Polynomial Problem Broyden10

δ		ICHEA+	I-ICHEA	ICHEA	imp
10^{-1}	SR	1.00	1.00	0.80	0.2
	Best	22 gens at 39.6s	31 gens at 45.0s	116 gens at 189.1s	4.8
	Median	29 gens at 55.8s	49 gens at 81.7	248 gens at 235.1s	4.2
	Worst	53 gens at 182.1s	158 gens at 300.0s	-	5.5
10^{-3}	SR	1.00	1.00	-	1.0
	Best	28 gens at 51.8s	36 gens at 59.4s	-	19.3
	Median	42 gens at 85.3s	38 gens at 74.0s	-	11.7
	Worst	79 gens at 269.3s	174 gens at 385.3s	-	3.7

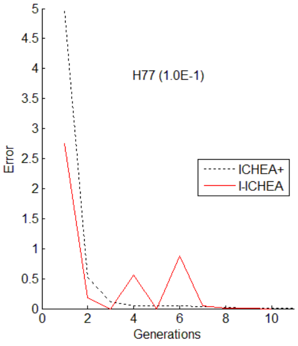


Fig. 2. I-ICHEA and ICHEA+ comparison for H77 ($\delta = 10^{-1}$)

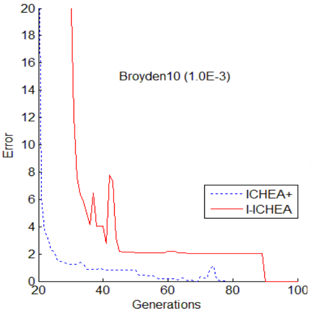


Fig. 3. I-ICHEA and ICHEA+ comparison for Broyden ($\delta = 10^{-3}$)

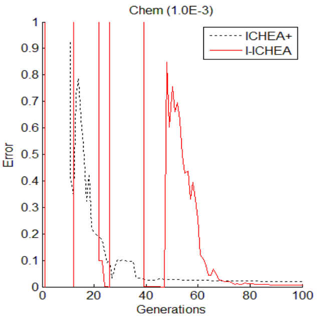


Fig. 4. I-ICHEA and ICHEA+ comparison for Chem ($\delta = 10^{-3}$)

$|\Omega|$ is the number of feasible solutions out of these $|pop|$ solutions. In the experimental setup, $|pop|$=10,000 and ρ value is computed as the average of five successive runs.

It has been demonstrated in [14] that ICHEA outperforms all other tested EAs where other tested EAs have very low SR. Hence we are only providing the results of ICHEA+ with previously introduced ICHEA. ICHEA has been developed in Java language and the tests have been carried out on the same Windows XP machine with Pentium (R) i5 CPU 2.52 GHz and 3.24 GB RAM. No parallel processing or distributed environment is used for the experiment. An average of 10 successive runs is taken into account to test the algorithms based on SR and generation count to reach to the solution. SR is the rate of successful trials for each problem i.e. $SR = successful\ trials\ /total\ trials$.

Nine test cases have been created using the benchmark problems from CSP domain [15] and COP domain [10, 11]. Tables 1, 3, 4, 5 show CSP test results for problems *Chem, HS109, Broyden10 and H77*, and Table 2 shows test results for a COP – *G05*. Each benchmark problem has been tested on two different δ values $\{10^{-1}, 10^{-3}\}$

Table 5. Benchmark trigonometry problem H77

δ		ICHEA+	I-ICHEA ($\rho\uparrow$)	I-ICHEA ($\rho\downarrow$)	ICHEA	imp
10^{-1}	SR	1.00	1.00	1.00	1.00	0.0
	Best	5 gens at 0.6s	6 gens at 0.7s	9 gens at 1.1s	8 gens at 0.3s	0.5
	Median	8 gens at 1.2s	6 gens at 0.7s	11 gens at 1.5s	22 gens at 0.64s	0.5
	Worst	13 gens at 2.0s	8 gens at 0.9s	13 gens at 2.0s	48 gens at 1.53s	0.8
10^{-3}	SR	1.00	1.00	1.00	1.00	0.0
	Best	7 gens at 0.91s	8 gens at 1.0s	20 gens at 3.4s	447 gens at 19.0s	20.9
	Median	16 gens at 2.3s	28 gens at 4.4s	41 gens at 6.8s	3250 gens at 113.7s	49.4
	Worst	21 gens at 3.1s	37 gens at 5.8s	66 gens at 11.3s	6297 gens at 211.4s	68.2

Table 6. Constraint Strengths for H77

Constraints	ρ
1	6.33E-01
2	6.14E-01
3	6.33E-04

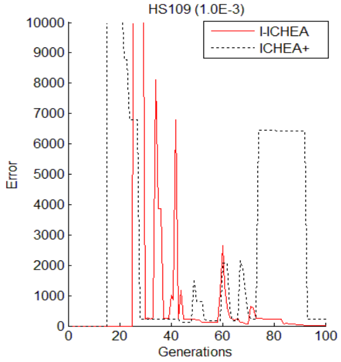


Fig. 5. I-ICHEA and ICHEA+ comparison for HS109 ($\delta = 10^{-3}$)

Table 7. Constraint Strengths for HS109

Constraints	1	2	3	4	5	6	7	8	9	10	11
ρ	0	0.87536	0.87662	0	0	0	0.00004	0.9241	0.44	0.41442	0.41874

except for problem *G05* which only uses $\delta = 10^{-5}$. Table 6 and Table 7 show constraint strengths for problems *H77* and *HS109* respectively. Other problems have same constraint strengths for all the constraints with $\rho = 0$. The ρ values are sorted in both ascending and descending order for separate tests where constraints are incrementally added to the search space in that order. As described earlier DCSPs are basically sequence of static constraints that are incrementally added to the search space. Table 1 – Table 5 show test results of ICHEA+ with previous ICHEA to compare their performances on different benchmark problems. The results of I-ICHEA (both ascending($\uparrow$) and descending($\downarrow$) order of ρ) have also been shown on the same tables to compare the results of ICHEA solving both static CSPs and dynamic CSPs as it is important to show whether knowledge from already solved constraints has been utilized or not. The test results are shown with best, median and worst solutions for each problem in terms of SR and efficiency. Columns are left blank with "-" if either it is not applicable or no good results have been obtained. The last column named *imp* shows the improvement of the ICHEA+ over ICHEA where the values for best, median and worst indicate how many times the ICHEA+ is better than ICHEA and the values for SR indicate the increase in SRs from ICHEA to ICHEA+.

Fig.2 – Fig. 5 depict the average of all test runs to compare performances of ICHEA+ and I-ICHEA. The y- axis shows the error value given in Eq. (8) and x-axis shows the number of generations. The graphical image shows the progress of ICHEA in solving CSP and DC SP. The spikes in the graphs for I-ICHEA indicate that a new constraint has been introduced into the search space and spikes for ICHEA+ indicate the current best individual at that generation has been improved by solving some additional constraints. This causes the fitness value of current best individual to increase as ICHEA+ favors individuals with less constraint violations which results in new individual (generally with high error value) to be added into the population [14].

6 Discussion

The experimental setup in Section 5 has dual objectives. Firstly, it demonstrates the comparative study of previously published ICHEA in [14] with an upgraded ICHEA that applies some new strategies proposed in Section 4.1 and Section 4.2 and secondly, whether ICHEA is able to handle dynamic constraints in an incremental manner by reusing knowledge from previous increments. The experimental results show that the addition of diversity management and stalled local optimal solutions management has improved the performance of ICHEA to solve CSPs. Previously introduced ICHEA has very low SR for many benchmark problems when δ is 10^{-3} because of local optimal solutions that makes the whole population become stagnant that has been massively improved for problems – *HS109*, *Broyden10* and *Chem*. ICHEA's efficiency has also been improved massively at different rates for all the hard problems except *G05* which is a simple problem in the perspective of CSPs. The second objective of the experiment is to show if I-ICHEA can perform similar to ICHEA+ where the experimental results show that I-ICHEA has not only performed similar to ICHEA+ but has outperformed it for problems – *HS109* and *Chem*. This demonstrates that ICHEA makes full use of constraints solved in previous increments that are transpired to new increments and it is capable of handling dynamic constraints. Constraints can be added dynamically to ICHEA and it can still give the solution with same efficiency and success as of solving all the constraints concurrently. The experimental results on the order of constraint strength did not produce any conclusive results about the performance of ICHEA as shown in problems – *H77* and *HS109* where results with mixed success have been observed.

7 Conclusion

This paper has proposed an improvement on ICHEA to solve CSPs together with an enhancement of its capacity to handle DCSPs effectively. It has been shown through benchmarks problems that the new strategies applied to ICHEA helps in maintaining the diversity of the populations and dealing with local optimal solutions by dynamically creating new constraints. This has helped massively in getting higher SR for most of the test problems. ICHEA has also been tested to handle DCSPs on benchmark CSPs that have been transformed to DCSPs. It has been shown that constraints can be added dynamically to ICHEA without restarting the algorithm and it can still give the solution with similar efficiency and SR as of solving all the constraints concurrently because ICHEA utilizes the knowledge from constraints of previous increments. The experimental results on the order of constraint strengths have been inconclusive in finding a CSP solution in an incremental approach. For future work efficiency of ICHEA can be tested on dynamic constraints where previous constraints can be removed or updated that distort the previous feasible regions. ICHEA has been able to solve CSPs and DCSPs. It has potential to be extended to work for discrete data as well because it extracts knowledge from constraints for its evolutionary search. ICHEA+ and I-ICHEA are currently further developed to solve real valued COPs and dynamic COPs respectively.

References

1. Brest, J., et al.: Dynamic optimization using Self-Adaptive Differential Evolution. In: IEEE Congress on Evolutionary Computation, CEC 2009, pp. 415–422 (2009)
2. Craenen, B.G.W., et al.: Comparing evolutionary algorithms on binary constraint satisfaction problems. IEEE Transactions on Evolutionary Computation 7, 424–444 (2003)
3. Deb, K., et al.: A fast and elitist multiobjective genetic algorithm. NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
4. Eiben, A.E.: Evolutionary Algorithms and Constraint Satisfaction: Definitions, Survey, Methodology, and Research Directions. In: Theoretical Aspects of Evolutionary Computing, pp. 13–58 (2001)
5. El Rhalibi, A., Kelleher, G.: An approach to dynamic vehicle routing, rescheduling and disruption metrics. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pp. 3613–3618 (2003)
6. Karimi, J., et al.: A new hybrid approach for dynamic continuous optimization problems. Applied Soft Computing 12, 1158–1167 (2012)
7. Kramer, O.: A Review of Constraint-Handling Techniques for Evolution Strategies. Applied Computational Intelligence and Soft Computing, 1–11 (2010)
8. Li, C., et al.: Benchmark Generator for CEC'2009 Competition on Dynamic Optimization (2008)
9. Li, T., et al.: Dynamic Constraint Satisfaction Approach to Hybrid Flowshop Rescheduling. In: 2007 IEEE International Conference on Automation and Logistics, pp. 818–823 (2007)
10. Liu, H., et al.: Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl. Soft Comput., 629–640 (2010)
11. Michalewicz, Z., Schoenauer, M.: Evolutionary algorithms for constrained parameter optimization problems. Evolutionary Computation 4, 1–32 (1996)
12. Nguyen, T., Yao, X.: Continuous Dynamic Constrained Optimisation - The Challenges. IEEE Transactions on Evolutionary Computation 99, 1 (2012)
13. Dechter, R.: Constraint networks. In: Encyclopedia of Artificial Intelligence, pp. 276–285. John Wiley & Sons, Ltd., New York (1992)

14. Sharma, A., Sharma, D.: ICHEA – A Constraint Guided Search for Improving Evolutionary Algorithms. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part I. LNCS, vol. 7663, pp. 269–279. Springer, Heidelberg (2012)
15. The COCONUT Benchmark, `http://www.mat.univie.ac.at/~neum/glopt/coconut/Bench-mark/Benchmark.html`
16. Tessema, B., Yen, G.G.: A Self Adaptive Penalty Function Based Algorithm for Constrained Optimization. In: IEEE Congress on Evolutionary Computation, pp. 246–253 (2006)
17. Verfaillie, G., Jussien, N.: Constraint Solving in Uncertain and Dynamic Environments: A Survey. Constraints, 253–281 (2005)
18. Wang, H., Wang, D.-W., Yang, S.: Triggered Memory-Based Swarm Optimization in Dynamic Environments. In: Giacobini, M. (ed.) EvoWorkshops 2007. LNCS, vol. 4448, pp. 637–646. Springer, Heidelberg (2007)

Constrained Multi-objective Optimization Using a Quantum Behaved Particle Swarm

Heyam Al-Baity[1,2], Souham Meshoul[3], and Ata Kaban[1]

[1] Computer Science Department, University of Birmingham, UK
{hha090,A.Kaban}@cs.bham.ac.uk
[2] Information Technology Department, King Saud University, Riyadh, Saudi Arabia
[3] Computer Science Department, MISC Laboratory, University Mentouri, Constantine, Algeria
smeshoul@umc.edu.dz

Abstract. The possibility to get a set of Pareto optimal solutions in a single run is one of the attracting and motivating features of using population based algorithms to solve optimization problems with multiple objectives. In this paper, constrained multi-objective problems are tackled using an extended quantum behaved particle swarm optimization. Two strategies to handle constraints are investigated. The first one is a death penalty strategy which discards infeasible solutions that are generated through iterations forcing the search process to explore only the feasible region. The second approach takes into account the infeasible solutions when computing the local attractors of particles and adopts a policy that achieves a balance between searching in infeasible and feasible regions. Several benchmark test problems have been used for assessment and validation. Experimental results show the ability of QPSO to handle constraints effectively in multi-objective context. However, none of the two investigated strategies has been found to be the best in all cases. The first strategy achieved the best results in terms of convergence and diversity for some test problems whereas the second strategy did the same for the others.

Keywords: Swarm Computing, Quantum behaved Particle Swarm Optimization, Constraint Handling, Multiobjective Optimization, Function Optimization.

1 Introduction

Quantum behaved Particle swarm optimization (QPSO) algorithm is a recent variant of PSO algorithm, proposed by Sun et al. [1] as a consequence of combining quantum mechanics principles and trajectory analysis of particle swarm optimization (PSO) algorithm. QPSO is characterized by its simplicity and easy implementation. Besides, it has better search ability and fewer parameters to set when compared against PSO [2]. Because of the quantum behavior, QPSO could perform well in finding the optimal solutions as it can improve the convergence capability of the global optimization [3]. Quantum behaved PSO has been proved efficient when compared with PSO [1, 3] and has been successfully applied to many optimization problems [4].

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 456–464, 2012.

Most real world multi-objective optimization problems involve linear and/or non linear constraints that can be of equality or inequality type. Generally, constrained optimization problems are difficult to solve. This is because finding a solution that satisfies all constraints is not an easy task. The constraints split the search space into two regions depending on the feasibility of solutions. The feasible region encompasses all solutions satisfying all constraints. Hence, it contains all solutions of the problem. In our work, we are interested in Pareto optimal solutions [5].The infeasible region contains solutions that violate at least one of the constraints. Constraints can be categorized as hard in which case they must be satisfied and soft where they may be satisfied to some extent [5].

There is a considerable number of methodologies found in the literature to solve constrained optimization problems with multiple objectives. One of these methodologies is to completely ignore infeasible solutions. Although it is simple, this approach may face difficulties in finding feasible solutions [5]. Penalty function methods are the most popular constraint handling techniques. In this method, penalty values are added to individuals violating the constraints [5]. Deb et al. [6] suggested a new idea to modify the definition of domination by turning it into constrained dominance of solutions by incorporating infeasible solutions during the search process.

Most of constraints handling methods for MOPs (CMOPs) proposed in the literature are used within evolutionary algorithms [7]. However, no in-depth study for handling constraints using swarm based algorithms like particle swarm optimization is available. In [8], Coello et al. proposed a simple scheme that has been used without a thorough investigation of how this impacts the search.

The aim of this work is twofold. First, we study the appropriateness of QPSO to deal with CMOPs and second we study the impact of discarding or taking into account infeasible solutions during the search process. The level at which constraint handling can be considered when extending QPSO is when the local attractor of each particle has to be computed. Therefore, two strategies to integrate constraint handling mechanism within a multiobjective QPSO are investigated. The first strategy discards infeasible solutions. According to this approach, only feasible solutions are generated through iterations and the search space is restricted only to the feasible region. The second strategy adopts a policy that balances the search between the feasible region and infeasible region. The previous two strategies have been chosen to study the effect of dealing with infeasible solutions in QPSO multiobjective constrained optimization problems.

The remainder of the paper is organized as follows: In section 2, we provide a description of QPSO algorithm. Section 3 provides a formal description of the tackled problem. Section 4 is devoted to the framework we propose to handle constrained multi-objective optimization. Section 5 reports on conducted experiments and obtained results. Finally, conclusions and perspectives are provided in section 6.

2 Description of QPSO

QPSO, a recent PSO like algorithm, is a probabilistic PSO algorithm proposed by Sun et al. [1]. It is inspired by the classical PSO method and quantum mechanics theories.

The traditional PSO uses the concept of classical mechanics in which a particle is depicted by its position and velocity. In the quantum mechanics, the particles are considered to lie in a potential field. The position of each particle is depicted by using a wave function $\Psi(\mathrm{x},\mathrm{t})$ instead of position and velocity. By using Monte Carlo simulation method, it has been found that the position x_i of a particle for dimension j is updated by the following equation [1][9] :

$$x_{ij}^{t+1} = p_{ij}^t \pm \beta.\left|mbest_j^t - x_{ij}^t\right|.\ln\left(1/u_{ij}^t\right) \quad for\, j = 1..D \tag{1}$$

Where, $\mathrm{u}_{\mathrm{ij}}^{\mathrm{t}}$ is a random number within the range [0,1] β is the Contraction expansion coefficient (CE). It is the only tunable parameter of QPSO and has a significant impact on controlling the convergence speed of the algorithm [2]. D is the problem dimension. $\mathrm{p}_{\mathrm{ij}}^{\mathrm{t}}$ is the local attractor of particle i and is evaluated by :

$$p_{ij}^t = \emptyset_{ij}^t\,.self\ best_{ij}^t + \left(1 - \emptyset_{ij}^t\right).global\ best_j^t\ for\, j = 1..D \tag{2}$$

$\emptyset_{ij}^t$ is a random number within the range (0,1). Finally, *mbest* is called the Mainstream Thought Point or the mean best position. It is the mean of self best positions of all particles and is evaluated by :

$$mbest_j^t = \frac{1}{N}\sum_{i=1}^{N} selfbest_{ij}\ for\, j = 1..D \tag{3}$$

where, *N* is the population size.

3 Problem Definition

A constrained multi-objective optimization problem can be formulated as follows [10]: Find the decision vector:

$$\overrightarrow{x^*} = (x_1{}^*, x_2{}^*, \dots, x_n{}^*)^T \in F \subseteq S \subseteq R^n$$

that optimizes (minimizes or maximizes) the set of *M* objectives

$$(f_1(\vec{x}), f_2(\vec{x}), \dots f_M(\vec{x}))$$

Where *F* is the feasible region that is delimited by the following inequality and equality constraints:

$$g_j(\vec{x}) \geq 0, \qquad j = 1,2,\dots,J$$
$$h_k(\vec{x}) = 0, \qquad k = 1,2,\dots,K$$

And *S* is the search space that is the part of the *n*-dimensional space R^n delimited by the lower and upper bounds of variables

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, \qquad i = 1,2,\dots,n$$

$x_i^{(L)} and\, x_i^{(U)}$ are the lower bound and the upper bound of decision variable x_i. When Pareto dominance [8] is used, finding the vector $\overrightarrow{x^*} = (x_1{}^*, x_2{}^*, \dots, x_n{}^*)^T$ that achieves the best compromise among the multiple objectives consists in determining the set of Pareto-optimal solutions. A vector u = (u_1, ... , u_k) is said to dominate vector v = (v_1, ... , v_k) denoted by (u $\lesssim$ v) if and only if (in the minimization case) :

$$\forall i \in \{1,2,..,k\}, \quad u_i \leq v_i \;\; and$$
$$\exists i \in \{1,2, \dots k\} : \; u_i < v_i$$

Therefore, the best solutions in the sense of Pareto dominance constitute the Pareto optimal set which is the set of nondominated solutions [5]. The representation of the Pareto optimal set in the objective space defines the Pareto optimal front [8, 11].

4 The Proposed Framework for CMOPs Using QPSO

One of the key issues when solving constrained optimization problems is how to implicate the infeasible solutions in the search process. In QPSO, as shown in equation (1), the computation of a particle position requires calculating its attractor which is a function of the self best position of the particle and the global best position recorded within the whole swarm as described by equation (2). Therefore, the local attractor is the level through which searching in the feasible and infeasible regions can be conducted. This fact is behind the idea we propose in this work. In our previous work [12], we suggested a framework to extend QPSO to unconstrained MOP. In this extension, a global best archive has been created to keep non-dominated solutions that are generated during the search process and we proposed a two level selection strategy that uses sigma values and crowding distance information in order to select the suitable guide for each particle. Our objective was to help convergence of each particle using sigma values while favoring less crowded regions in the objective space to attain a uniformly spread out Pareto front. In this work, we further extend this framework to constrained MOPs by investigating two strategies to deal with infeasible solutions generated during the search process.

4.1 First Strategy

The first strategy consists simply in discarding infeasible solutions and using only the feasible ones throughout the search process. In this way, the whole swarm is forced to move within the feasible region *F*. Therefore, only feasible solutions are used to update local attractors of particles.

4.2 Second Strategy

In the second strategy, we suggest to explore both feasible and infeasible regions using a selection rule of global best position and self best position.

Global Best Position Selection Rule: Best infeasible solutions encountered during the search process are kept within an archive that we denote by Global Best Infeasible Archive (GBIA). The best infeasible solution is the one with lowest constraints violation and/or better objective values in terms of dominance relation. Given a particle for which a new position has to be computed, the global best solution for this particle is selected either from the Global best feasible archive (GBFA) or the infeasible archive (GBIA) according to a probabilistic rule as follows:

```
$p = \text{rand}(\,)$;
**If** ($p < P_G$) **Then** select Globalbest position from GBIA
**Else** select Globalbest position from GBFA ;
```

P_G is the selection probability and can be set according to the importance we wish to devote to infeasible solutions. Selecting from $GBIA$ is straightforward. It consists in choosing the global leader randomly as all infeasible solutions in the GBIA have the same quality measure in terms of number of constraint violation. Selecting from GBFA follows the same principle we described in [12]. The decision about which GBFA member to select as a leader for a given particle is made based on the closeness of each GBFA member to the current particle in terms of sigma values and the extent to which the local area around the member is crowded. The aim is to help convergence to the Pareto optimal front while ensuring a uniformly spread out front. The selection mechanism starts by identifying the k nearest GBFA members to the current particle using their sigma values. Then the less crowded member among these k neighbors is chosen as the global best solution used to compute the attractor of the current particle. More formally, the proposed selection method can be described as follows:

```
**Selection_Method** (GBFA, Current-Particle)

    Compute_Sigma_value (Current-Particle );
    **Foreach** ($M \in GBFA$ )
        Compute_Sigma_value ($M$ );
    **End foreach**
    Record_ k _nearest _neighbors;
    Choose_less_crowded_solution;

**End.**
```

As described in [13, 12], a sigma value of a particle P_i characterizes the line joining the corresponding point in the objective space to the center point (0,0,…,0). The closeness of two sigma values is in fact an indication that the two corresponding particles lie on two lines that are close to each other. This fact is used to guide the particle by the suitable leader. That is why the k nearest neighbors are selected according to the ascending order of the distance between the particle sigma value and each of the GBFA member sigma value. Crowding distance computation is done in a similar way as in [6]. Solutions in the GBFA are first sorted in the objective space then the overall crowding distance is calculated as the sum of individual distance values corresponding to each objective.

Both archives, GBFA and GBIA need to be updated after computing all particles' new positions. For a current particle, if a new infeasible solution is derived, its insertion in GBIA is considered. This new infeasible position enters the GBIA archive only if it has less constraint violation than any infeasible solution in GBIA. In this case, all GBIA contents with higher constraint violation have to be deleted from the archive. The new infeasible position is also included in case it is equal with all GBIA solutions in terms of number of constraint violation and dominance criteria. By another side, if a new feasible solution is derived, it has to be checked against GBFA contents. The comparison here is based on the usual dominance concept. The new feasible position is entered into GBFA in a way that keeps GBFA domination free.

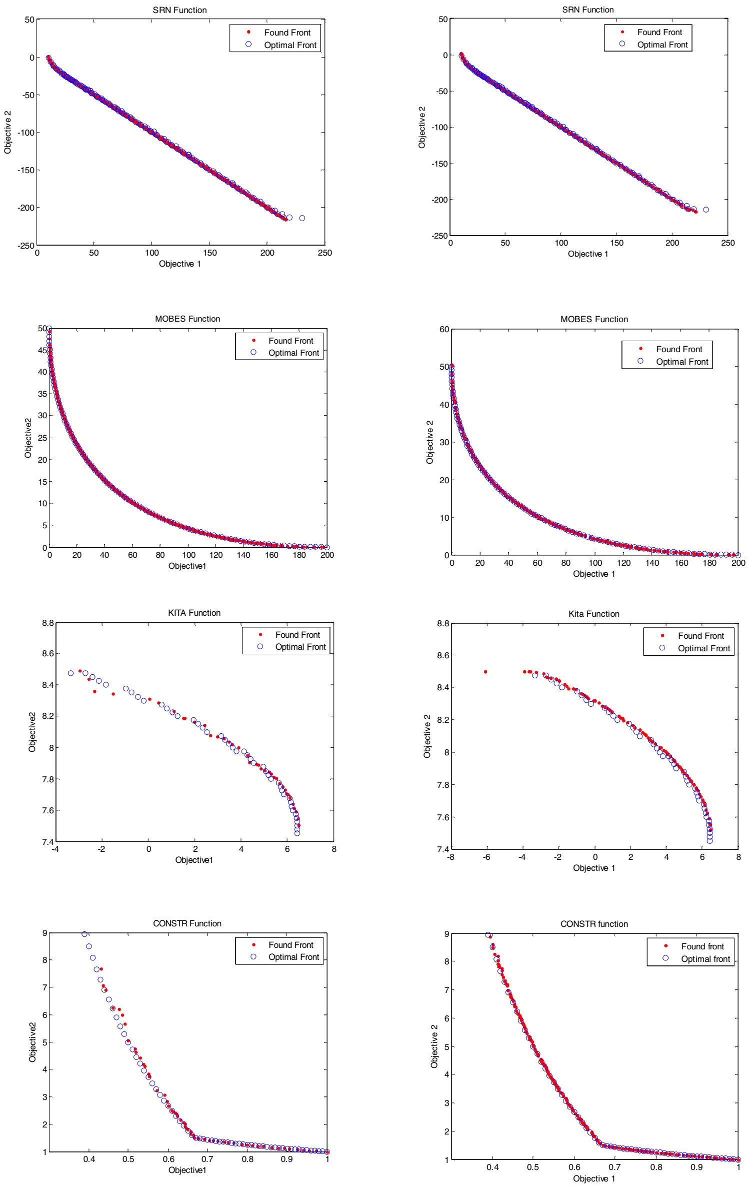
Self Best Position Selection Rule: Basically, the strategy we followed in our previous work [12] was to keep only one solution as a self best feasible (SBF) point for each particle. In this work, we keep track of the self best infeasible solution (SBI) for each particle as well. This self best infeasible solution has to be updated whenever another better infeasible individual is encountered. The SBI solution is the one with lowest constraint violation. When a new position has to be computed for a particle, the self best solution for this particle is selected either as the self best feasible (SBF) solution or the self best infeasible (SBI) solution according to a probabilistic rule as follows:

```
$p = \text{rand}( );$
If ($p < P_s$) Then select $SBI$ as the self best position
Else select $SBF$ as the self best position
```

P_s is the selection probability and can be set according to the importance we wish to devote to infeasible solutions. Finally, once self best and global best positions are determined for a particular particle, the local attractor can be computed as given in equation (3).

5 Experimental Results

Several experiments have been conducted to assess the performance of the QPSO for CMOPs using the two constraint handling strategies described above. The test problems used for this purpose can be found in [6, 8, 10]. In order to evaluate the performance of both strategies in terms of convergence and diversity of the obtained fronts, two metrics have been used namely the Generational Distance (GD) and the Spacing metric (SP) described in [8]. In all experiments, the contraction expansion



(a) (b)

Fig. 1. Optimal front and obtained front for the four test functions (a) first strategy (b) second strategy

parameter β has been decreased linearly within the range [1.2-0.5]. The number k of neighbors in the selection of the global feasible leader has been set to 10 and selection probabilities P_G and P_S to 0.5. The maximum number of iterations has been set to 700 for KITA function and 500 for the other functions. Figure 1 shows the obtained fronts along with the optimal fronts using strategy 1 and strategy 2 respectively for SRN, MOBES, KITA and CONSTR test functions. At a first glance, we can see that very good convergence and diversity have been achieved in both cases. To corroborate this fact, quantitatively speaking, ten runs in each strategy for each function have been conducted. Statistics have been gathered in Table 1. The values of the standard deviation show the robustness and the high quality of the found solutions. Regarding the two investigated strategies, it is apparent that discarding infeasible solutions during the search process has led to the best results from both convergence and diversity points of view in case of SRN and MOBES functions whereas the second strategy is more efficient in case of KITA and CONSTR functions.

Table 1. Metrics' values for first and second strategy

	Statistics	First Strategy		Second Strategy	
Test problem		GD	SP	GD	SP
SRN function[6]	Best	**0.0217**	**0.00009**	0.0269	0.0007
	Worst	**0.0223**	**0.0010**	0.0280	0.0135
	Average	**0.0220**	**0.0007**	0.0277	0.0042
	Median	**0.0219**	**0.0010**	0.0279	0.0013
	Std. Deviation	**0.0002**	**0.0004**	0.0005	0.0062
CONSTR function[6]	Best	0.0063	0.0038	**0.0049**	**0.0004**
	Worst	0.0089	0.0124	**0.0052**	**0.0010**
	Average	0.0075	0.0060	**0.0051**	**0.0007**
	Median	0.0074	0.0039	**0.0051**	**0.0007**
	Std. Deviation	0.0009	0.0037	**0.0002**	**0.0004**
KITA function[8]	Best	0.0217	0.0425	**0.0143**	**0.0008**
	Worst	0.4205	1.6729	**0.1720**	**0.0247**
	Average	0.2175	0.4566	**0.0716**	**0.0096**
	Median	0.1969	0.0634	**0.0668**	**0.0084**
	Std. Deviation	0.1500	0.7013	**0.0543**	**0.0077**
MOBES function[10]	Best	**0.0194**	**0.0017**	0.0296	0.0021
	Worst	**0.0201**	**0.0860**	0.0338	0.0106
	Average	**0.0197**	**0.0312**	0.0310	0.0057
	Median	**0.0197**	**0.0030**	0.0299	0.0045
	Std. Deviation	**0.0002**	**0.0404**	0.0019	0.0036

6 Conclusion

In this paper we investigated the use of QPSO to handle CMOPs. Two strategies to deal with infeasible solutions have been studied that consist in discarding versus taking into account infeasible solutions. In both cases, the extended QPSO has been successfully applied to CMOPs. However, none of the two strategies has been found to achieve the best results in terms of convergence and diversity in all cases.

References

1. Sun, J., Feng, B., Xu, W.: Particle Swarm Optimization with Particles having Quantum Behavior. In: IEEE Proceedings of Congress on Evolutionary Computation, pp. 325–331 (2004)
2. Fang, W., Sun, J., Ding, Y., Wu, X., Xu, W.: A review of Quantum-behaved Particle Swarm Optimization. IETE Technical Review (2010)
3. Sun, J., Xu, W., Feng, B.: A Global Search Strategy of Quantum-behaved Particle Swarm Optimization. In: IEEE Conference on Cybernetics and Intelligent Systems, pp. 111–116 (2004)
4. Meshoul, S., Al-Owaisheq, T.: QPSO-MD: A Quantum Behaved Particle Swarm Optimization for Consensus Pattern Identification. In: Cai, Z., Li, Z., Kang, Z., Liu, Y. (eds.) ISICA 2009. CCIS, vol. 51, pp. 369–378. Springer, Heidelberg (2009)
5. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, Chichester (2001)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGAII. IEEE Transactions on Evolutionary Computation, 182–197 (2002)
7. Coello, C.A.: Constraint-Handling using an Evolutionary Multiobjective Optimization Technique. Civil Engineering and Environmental Systems 17, 319–346 (2000)
8. Coello, C.A., Pulido, G.T., Lechuga, M.S.: Handling multiple objectives with particle swarm optimization. IEEE Transactions on Evolutionary Computation, 256–279 (2004)
9. Sun, J., Lai, C.H., Xu, W.-B., Chai, Z.: A Novel and More Efficient Search Strategy of Quantum-Behaved Particle Swarm Optimization. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 394–403. Springer, Heidelberg (2007)
10. Binh, T., Korn, U.: MOBES: A Multiobjective Evolution Strategy For Constrained Optimization Problems. In: Proceedings of the Third International Conference on Genetic Algorithms (Mendel 1997), pp. 76–182 (1997)
11. Abranham, A., Jain, L.: Evolutionary multiobjective optimization. In: Ajith, A., Lakhmi, J., Robert, G. (eds.) Evolutionary Multiobjective Optimization, Advanced Information and Knowledge Processing, pp. 1–6 (2005)
12. AlBaity, H., Meshoul, S., Kaban, A.: On Extending Quantum Behaved Particle Swarm Optimization to MultiObjective Context. In: Proceedings of the IEEE World Congress on Computational Intelligence (IEEE CEC 2012), pp. 996–1003 (2012)
13. Mostaghim, S., Teich, J.: Strategies for finding good local guides in multi-objective particle swarm optimization (mopso). In: Proceedings of the IEEE Swarm Intelligence Symposium, pp. 26–33 (2003)

Learning from Positive and Unlabelled Examples Using Maximum Margin Clustering

Sneha Chaudhari[1,⋆] and Shirish Shevade[2,⋆⋆]

[1] IBM Research Lab, Bangalore, India
snechaud@in.ibm.com
[2] Indian Institute of Science, Bangalore, India
shirish@csa.iisc.ernet.in

Abstract. Learning from Positive and Unlabelled examples (LPU) has emerged as an important problem in data mining and information retrieval applications. Existing techniques are not ideally suited for real world scenarios where the datasets are linearly inseparable, as they either build linear classifiers or the non-linear classifiers fail to achieve the desired performance. In this work, we propose to extend maximum margin clustering ideas and present an iterative procedure to design a non-linear classifier for LPU. In particular, we build a least squares support vector classifier, suitable for handling this problem due to symmetry of its loss function. Further, we present techniques for appropriately initializing the labels of unlabelled examples and for enforcing the ratio of positive to negative examples while obtaining these labels. Experiments on real-world datasets demonstrate that the non-linear classifier designed using the proposed approach gives significantly better generalization performance than the existing relevant approaches for LPU.

Keywords: Learning from Positive and Unlabelled Examples, Maximum Margin Clustering, Least Squares Support Vector Classifier.

1 Introduction

Many applications of information retrieval and data mining face binary classification problems which typically involve datasets consisting of a small set of positive examples and a large number of unlabelled examples. This problem of Learning from Positive and Unlabelled examples (LPU) occurs in situations where either characterizing negative examples is difficult or their annotation is expensive. Consider the real world application of Junk Mail Filtering [1]. Here, the junk messages serve as positive examples as they can be distinguished from legitimate mails in terms of style and vocabulary; they are independent of individual users and, hence, easier to characterize and annotate. Consequently, the

[⋆] This work was done while the author was a graduate student in the Dept of CSA, IISc, India.

[⋆⋆] The author gratefully acknowledges the support for this work from Infosys Ltd, India.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 465–473, 2012.

aim is to learn to filter junk mails automatically to improve the usability of an e-mail client.

Motivation and Related Work: Many of the existing approaches for handling the problem of LPU [2], [3] construct a linear classifier. These approaches do not achieve the desired performance for some real world scenarios, as linear classifiers are not sufficient where the datasets are linearly inseparable. To remedy this, Support Vector Machines (SVM) based approaches have been proposed which can obtain a non-linear classifier by employing a kernel function. However, as observed in [4], SVM based approaches suffer from the risk of premature convergence due to the asymmetry of the hinge loss function of SVMs. Further, existing techniques do not enforce the class balance ratio, i.e., ratio of positive to negative examples in the unlabelled data, which is useful for avoiding trivial solutions and obtaining better generalization performance.

For example, consider a one-class SVM proposed in [5] which uses only positive examples for learning, resulting in poor performance. Further, iterative SVM based approaches have been proposed where the final classifier is either the last classifier obtained after convergence [6], or a selected classifier from the set of classifiers built [7]. However, for training the SVM, these methods obtain the labels of unlabelled examples using different techniques. A cost asymmetric SVM formulation, called Biased-SVM (BSVM) is proposed in [3]. BSVM uses two parameters to assign a higher weight to positive errors in comparison to negative errors. Further, it uses the Naive Bayes (NB) classifier for initializing the labels of unlabelled examples. One more approach based on similar ideas of BSVM method is given in [8], where a probabilistic approach is followed to assign the weights to positive and unlabeled examples. Another interesting approach is presented in [9], where a Positive Naive Bayes (PNB) classifier is constructed by adapting the NB classifier to handle the problem of LPU. Recently, a practical approach for Maximum Margin Clustering (MMC) has been proposed in [4]. MMC performs clustering by finding a decision surface passing through low density region in the data. The optimization problem in MMC is non-convex and an iterative procedure is adopted in [4], using Support Vector Regression with Laplacian loss to avoid premature convergence.

Contributions: In this work, we extend the idea of iterative learning adopted in [4] to the problem of LPU and design a *non-linear* classifier. The classifier is designed using Least Squares SVM (LS-SVM) [10] method. It effectively handles the non-convexity of the optimization problem involved, by virtue of a symmetric loss function. Positive examples and the class balance ratio are used to determine the bias term in the classification model. This helps to avoid trivial solutions and improve the performance on unseen data. As the class balance ratio is not exactly known in practice, we experimentally show that the proposed approach is useful even if the value is approximately known. Further, appropriate initialization of the labels of unlabelled examples is crucial in this problem set-up and we propose a simple technique for this purpose, which is effective in improving the performance. Though maximum margin classification ideas have been used in past for semi-supervised learning, to the best of our knowledge, *Maximum Margin Clustering* has not been explored before to handle the problem of LPU. Experimental results on real-world datasets demonstrate that the proposed approach is useful for designing a non linear classifier with significantly improved generalization performance than existing techniques such as Iterative SVM (ISVM), BSVM and PNB.

2 Proposed Approach: Maximum Margin Clustering with Least Squares SVM (MCLS)

The problem of learning from positive and unlabelled training examples is to obtain a binary classifier, given a training set consisting of N examples, where the first L examples, $\{x_i, +1\}_{i=1}^{L}$ are positive and the remaining $U = N-L$ examples, $\{x_i\}_{i=L+1}^{N}$, are unlabelled. In this work, we design a non-linear support vector classifier of the form f(x) = $w^\top \varphi(x) + b$, where $\varphi(x)$ is a non-linear function. The underlying optimization problem is given in (1).

$$\begin{aligned} \min_{w,b,\{y_i\}_{i=L+1}^{N}} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{L} l(w^T\varphi(x_i)+b) + \sum_{i=L+1}^{N} l(y_i(w^T\varphi(x_i)+b)) \\ s.t. \quad & y_i \in \{+1,-1\} \quad \forall i = L+1 \longrightarrow N \\ & \frac{1}{U}\sum_{i=L+1}^{N} max(0, sign(w^T\varphi(x_i)+b)) = r \end{aligned} \tag{1}$$

where, C is a positive hyper-parameter which controls the trade-off between smoothness and fitness and $l(t)$ is a loss function; for example, the hinge loss function in SVMs is $l(t) = max(0, 1-t)$. As we can see, this optimization problem is a variant of Transductive SVM formulation [11], which introduces separate terms in the objective function for positive and unlabelled examples. Also, notice that it is important to add the second constraint, which specifies that a fraction r of unlabelled data is to be labelled positive. This user defined parameter ensures that the class balance ratio is maintained in the set of unlabelled examples.

This non-convex optimization problem (1) is hard to solve. Hence, we employ a practical approach to obtain a solution to LPU. The proposed approach adopts an iterative procedure that learns a non-linear LS-SVM classifier. In each iteration, we first fix the labels of unlabelled examples and optimize with respect to w and consequently, fix w and find new labels of unlabelled data. Precisely, the proposed approach consists of the following main steps : (i) We initialize the labels of unlabelled data using the algorithm explained in Subsection 2.1. (ii) To optimize with respect to w, we train LS-SVM classifier using a labelled training set obtained in (i). We make use of LS-SVM as a classification algorithm as it avoids poor local minima due to a symmetric loss function. We explain this in detail in Subsection 2.2. (iii) We determine the new labels of unlabelled data using the decision function of the LS-SVM classifier. However, these labels are computed such that the second constraint in (1) is satisfied. The proposed approach maintains r by appropriately determining the bias parameter, b. The procedure is described in Subsection 2.3. Finally, these steps are repeated until the labels of unlabelled examples remain constant in successive iterations or maximum nunber of iterations is reached. This procedure is given succinctly in Algorithm 1. Now we discuss each aspect of the method in detail in the following subsections.

2.1 Initialization of Labels of Unlabelled Data (ILU)

The initialization of labels of unlabelled examples (step 1 in Algorithm 1) is very crucial as they are used to train the LS-SVM. We propose a method for obtaining these labels which is effective in improving the accuracy. Initially, k-means clustering is performed on the training data. Each cluster is determined as positive or negative, according to the number of positive examples present in that cluster. To obtain negative examples, some examples are chosen from each of the negative clusters which are farthest from the centroid of positive examples. The intuition is to select those examples from the unlabelled data, which have higher probability of belonging to the negative class. The number of examples to be selected depends on the value r of the dataset. Now, any supervised classification technique can be used for training where the input is the set of positive examples and the selected negative examples; we use SVM in our algorithm. Finally, the labels of all the unlabelled examples are obtained using the decision function of the classifier.

2.2 Non-linear LS-SVM Classifier

The LS-SVM formulation for a completely labelled dataset $\{x_i, y_i\}_{i=1}^N$ can be given as follows:

$$\begin{aligned} \min_{w,b,\xi_i} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i^2 \\ \text{s.t.} \quad & y_i - (w^\top \varphi(x_i) + b) = \xi_i \quad \forall i \end{aligned} \tag{2}$$

The main benefit of LS-SVM classifier is that it is well suited for solving this problem due to symmetry of its loss function [4]. The loss remains the same, if the label of an unlabelled example is changed during training. This encourages necessary flipping of labels and classifier improves over the initial labels. This in turn helps to avoid many poor local minima and obtain a better solution.

2.3 Maintaining Class Balance Ratio

Maintaining the class balance ratio, r in the labels of unlabelled data (step 4 in Algorithm 1) is necessary to avoid trivial solutions such as assigning all examples to one class to obtain an unbounded margin hyperplane. The proposed algorithm performs a simple, efficient and easy to implement computation of the bias value (b) of LS-SVM to maintain this ratio. At the same time, the algorithm also tries to set b such that the labels of positive examples remain constant while finding the labels of unlabelled examples, a critical necessity for applications of LPU. The algorithm uses r and a tolerance parameter which decides the trade-off between maintaining the class balance and correctly classifying the positive examples. The algorithm sorts $w^T\varphi(x)$ values and sets the bias value to the $w^T\varphi(x)$ value satisfying r. The algorithm now checks if all the positive examples are correctly classified. Otherwise, the bias is changed in the range of the tolerance parameter such that maximum number of positive examples are correctly classified.

Algorithm 1 MCLS

```
Input: Training set $\{x_i, +1\}_{i=1}^{L} \cup \{x_i\}_{i=L+1}^{N}$, where $y_i \in \{+1, -1\}, \forall i = L+1 \longrightarrow N$
Output: Classifier : f(x) = $w^\top\varphi(x) + b$
Find labels of unlabelled examples ($\{\bar{y_i}\}_{i=L+1}^{N}$) using algorithm described in 2.1
while TRUE do
    Perform LS-SVM training using $\{x_i, +1\}_{i=1}^{L} \cup \{x_i, \bar{y_i}\}_{i=L+1}^{N}$ and compute $w$
    Compute the bias value ($\hat{b}$) using the method described in 2.3
    Obtain new labels: $\hat{y_i}$ using $w$ and bias value $\hat{b}$
    i.e. $\hat{y_i} = \text{sign}(w^\top\varphi(x_i) + \hat{b}) \quad \forall i = 1 \longrightarrow N$
    if $\bar{y_i} == \hat{y_i}\ \forall i = 1 \longrightarrow N$ then
        Break
    else
        $\bar{y_i} = \hat{y_i}\ \forall i = 1 \longrightarrow N$
    end if
end while
$b = \hat{b}$
```

3 Experimental Evaluation

The experimental study was conducted on seven real world datasets, as given in Table 1. The six datasets in Table 1 except ionosphere are available at http://theoval.cmp.uea.ac.uk/∼gcc/matlab/default.html#benchmarks. The ionosphere dataset has been taken from the UCI machine learning repository [12]. MCLS, Naive Bayes (NB), Iterative SVM (ISVM) and Positive Naive Bayes (PNB) were implemented in Matlab (version R2010a). For all the experiments, we used RBF kernel function defined as : $K(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$. Also, the generalization performance of the classifiers designed using different techniques was studied as we increased the number of positive examples (L). In particular, we chose 5, 10, 15, 20 and 25% of actual positive class examples in the training set.

Demonstration of MCLS on a Toy Dataset: We consider a two dimensional toy dataset to demonstrate the decision boundaries obtained by MCLS. The dataset consists of 400 examples with $r = 0.5$. Figure 1(a) shows the decision boundary obtained using a completely labelled training set. The rest of the

Table 1. Details of Datasets. N: Total number of examples, TR: Number of training examples, TS: Number of test set examples, **r**: Class balance ratio, ATS: Accuracy on test set

Dataset	**N**	**TR**	**TS**	**r**	**ATS(%)**
banana	5300	3533	1767	0.4483	90.26
thyroid	215	143	72	0.3	97.22
heart	270	180	90	0.4444	83.22
pima	768	512	256	0.349	74.6
waveform	5000	3333	1667	0.3294	90.53
ringnorm	7400	4934	2466	0.495	98.78
ionosphere	351	234	117	0.641	96.583

plots, Figures 1(b)-1(f), show the decision boundaries given by MCLS for different values of L. The plots clearly show the efficacy of the proposed algorithm. In particular, for $L = 15\%$ (Figure 1(d)), the decision boundary is very close to the one obtained using the completely labelled data. We also show decision boundaries given by MCLS and other techniques in Figure 2 with L=15%. Note that MCLS obtains a reasonable decision boundary than other existing techniques, when only 15% positively labelled examples are used.

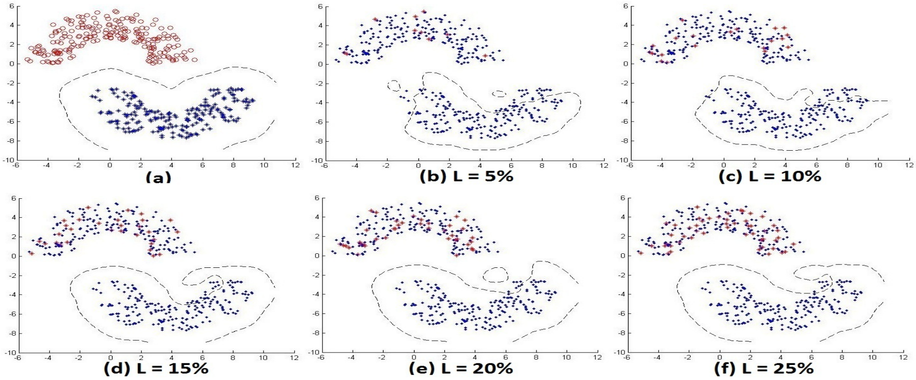

(a)
(b) L = 5%
(c) L = 10%
(d) L = 15%
(e) L = 20%
(f) L = 25%

Fig. 1. Decision boundary obtained by MCLS algorithm as L increases. Positive and Unlabelled examples are shown by red stars and blue dots respectively. (a) Shows the decision boundary obtained using labelled training set.

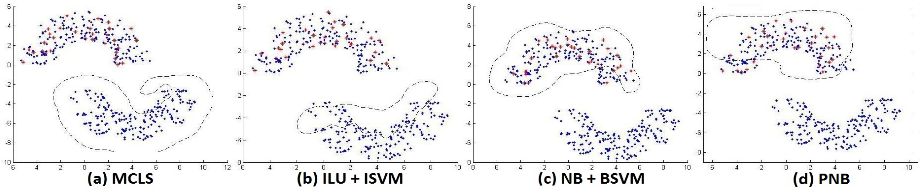

(a) MCLS
(b) ILU + ISVM
(c) NB + BSVM
(d) PNB

Fig. 2. Decision boundary obtained by MCLS and existing techniques with L=15%

Generalization Performance of MCLS: In Table 2, we report the test set accuracies of MCLS as a function of the number of positive examples, compared with following methods (1) ILU (Subsection 2.1) + ISVM [6]. Here, after initialization using ILU, SVM is trained iteratively and the last classifier obtained after convergence is selected. (2) NB + BSVM [3] and (3) PNB [9]. MCLS shows significantly better accuracies for all datasets when compared to the rest of the three algorithms. The iterative SVM does not perform well as it faces the problem of getting stuck in poor local minima. The BSVM method, though assigns different weights to positive and unlabelled examples, does not focus on maintaining the r fraction in the labels of unlabelled data. The PNB method does not show comparable performance. Further, the difference in the accuracies is prominent for small values of L. This demonstrates the applicability of MCLS for datasets with small number of positive examples. For the datasets heart, pima, ionosphere and banana, the performance using $L = 25\%$ is comparable with that obtained using a completely labelled training set.

Performance Evaluation of ILU: To evaluate the algorithm described in Subsection 2.1, we compared the accuracies obtained over unlabelled data with one popular approach proposed in [3] for initialization of labels. The authors construct a NB Classifier by treating all unlabelled examples as negative. The results are given in Table 3. ILU outperforms NB on almost all the datasets. Also, ILU shows greater increase in the accuracy as we increase L compared to NB. The reason is that NB is constructed by treating all unlabelled examples as negative whereas ILU algorithm constructs SVM classifier by extracting negative examples from unlabelled examples.

Table 2. Comparison of Test set Accuracies of MCLS, Iterative SVM (ISVM), Biased SVM (BSVM), Positive Naive Bayes (PNB) Algorithms

L	10%				15%				20%				25%			
Dataset	MCLS	ILU + ISVM	NB + BSVM	PNB	MCLS	ILU + ISVM	NB + BSVM	PNB	MCLS	ILU + ISVM	NB + BSVM	PNB	MCLS	ILU + ISVM	NB + BSVM	PNB
banana	**80.9**	72	70	63.7	**81.8**	73.1	75.3	67.6	**84.6**	77.5	81.4	68.9	**87.4**	82.1	85.7	70.5
thyroid	81.9	**84.2**	80.5	74.4	86.6	**87.3**	86.1	76.3	**90.7**	90.3	87.5	79.2	**93**	91.6	91	80.9
pima	**67.9**	64.8	64.4	60.9	**68.3**	66.7	67.5	65.1	**71.4**	67.8	68.1	66.4	**73**	68.7	69.9	69.9
ionosphere	**82.3**	75.2	78.6	71.8	**88**	76.3	77.7	76.1	**90.5**	80.3	85.4	79.4	**94**	81.2	90.4	83.7
heart	**72.1**	67.7	67.9	63.2	**73.3**	68.8	69.9	66.5	**78.1**	74.1	73.6	70.7	**81.4**	77.8	78	72.2
waveform	**78.3**	75.1	70.4	70.8	**79.4**	75.9	70.8	72.6	**82.4**	77.3	70.9	74.2	**82.7**	78.3	71.1	74.9
ringnorm	**92.5**	87.7	88.6	87	**92.6**	89.6	89.5	88.2	**94.4**	90.9	91.4	89.7	**94.8**	91.1	92	90.2

Table 3. Comparison of Accuracies over unlabelled data of ILU and NB

L	**5%**		**10%**		**15%**		**20%**		**25%**	
Dataset	**ILU**	**NB**	**ILU**	**NB**	**ILU**	**NB**	**ILU**	**NB**	**ILU**	**NB**
banana	**72.3**	54.5	**73.8**	59.3	**74**	62.5	**77.1**	64.8	**77.6**	64.2
pima	**66.1**	58.3	**66**	59.3	**66.9**	58.9	**67.7**	60.4	**66.9**	60.7
heart	**68.3**	51.3	**69.5**	60.4	**70.6**	58.3	**73.9**	62	**77.6**	67.7
ionosphere	**67.3**	55.6	**72.3**	65.3	**74.2**	65.6	**75.8**	63.9	**76.4**	69.5
ringnorm	**90**	86	**90.5**	85	**90.4**	87.1	**91**	87.4	**91.1**	87.6
thyroid	**78.6**	58.6	**79.7**	73.2	**83.4**	82.3	**88.2**	86	**89.1**	86.7
waveform	**76.3**	74.6	**78.2**	76.1	**78.8**	77.4	**79.8**	78.5	77.3	**79.7**

Variation of r Fraction: The parameter r (fraction of positive examples in unlabeled data) is typically not exactly known in practice. We therefore conducted an experiment to study the generalization performance of the classifier designed using the proposed method, when r is varied in a small interval around its true value. The results are reported in Table 4 for four datasets. It is evident from this table that is no significant degradation in the generalization performance in small neighborhood of r. Thus, the proposed approach is useful even if the value of parameter r is approximately known.

Table 4. Test set Accuracy of MCLS as a function of parameter r

Dataset	**r**	**r-0.15**	**r-0.1**	**r-0.05**	**r+0.05**	**r+0.1**	**r+0.15**
banana	**84.6**	79.7	81	82.5	81.8	80.7	78.4
pima	**71.4**	69.2	70.7	71	69.1	68.7	67.5
heart	**78.1**	71.1	73.3	75.5	75.5	72.2	68.8
thyroid	**90.7**	86.1	86.8	87.5	88.8	87.5	83.3

4 Conclusion

In this work, we consider the problem of learning from positive and unlabelled examples by proposing a new approach to build a Least Squares support vector classifier, based on Maximum Margin Clustering. The proposed approach is particularly useful for real world applications where there is necessity of non-linear classifiers with good generalization performance. The proposed method gives significantly better accuracy than exiting techniques, especially with small number of positive examples. We also performed experiments with different values of r in the range of $r\pm0.15$. The proposed approach showed minor degradation in the performance as r was varied in the specified range. Thus, the proposed approach is an useful alternative for learning from positive and unlabelled examples.

References

1. Schneider, K.-M.: Learning to Filter Junk E-Mail from Positive and Unlabeled Examples. In: Su, K.-Y., Tsujii, J., Lee, J.-H., Kwong, O.Y. (eds.) IJCNLP 2004. LNCS (LNAI), vol. 3248, pp. 426–435. Springer, Heidelberg (2005)
2. Zhang, B., Zuo, W.: Learning from Positive and Unlabeled Examples: A Survey. In: Yu, F., Luo, Q. (eds.) International Symposium on Information Processing, pp. 650–654. IEEE Computer Society (2008)
3. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building Text Classifiers Using Positive and Unlabeled Examples. In: Proceedings of the 3rd IEEE International Conference on Data Mining, pp. 179–188 (2003)
4. Zhang, K., Tsang, I.W., Kwok, J.T.: Maximum Margin Clustering Made Practical. IEEE Transactions on Neural Networks 20(4), 583–596 (2009)
5. Manevitz, L.M., Yousef, M.: One-class SVMs for Document Classification. Journal of Machine Learning Research 2, 139–154 (2001)
6. Zhang, B., Zuo, W.: Reliable Negative Extracting Based on kNN for Learning from Positive and Unlabeled Examples. Journal of Computers 4(1), 94–101 (2009)

7. Yu, H., Han, J., Chang, K.C.C.: PEBL: Positive Example Based Learning for Web Page Classification using SVM. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 239–248. ACM Press, New York (2002)
8. Elkan, C., Noto, K.: Learning Classifiers from Only Positive and Unlabeled Data. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 213–220. ACM, New York (2008)
9. Calvo, B., Larraaga, P., Lozano, J.A.: Learning Bayesian Classifiers from Positive and Unlabeled Examples. Pattern Recognition Letters 28(16), 2375–2384 (2007)
10. Suykens, J.A.K., Vandewalle, J.: Least Squares Support Vector Machine Classifiers. Neural Processing Letters 9, 293–300 (1999)
11. Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines. In: Proceedings of the Sixteenth International Conference on Machine Learning, pp. 200–209. Morgan Kaufmann Publishers Inc., San Francisco (1999)
12. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2007)

Novel Robust Stability Criteria for Stochastic Hopfield Neural Network with Time-Varying Delays

Xiaolin Li and Minrui Wang

Department of Mathematics, Shanghai University, Shanghai 200444, China
{xlli,Incwxmxr}@shu.edu.cn

Abstract. In this paper, stochastic Hopfield neural networks with time-varying delays are investigated based on Lyapunov-krasovskii functional approach and linear matrix inequality(LMI) technique. The proposed criterion is expressed in terms of linear matrix inequality(LMI)and is less conservative than some existing ones and can be effectively solved by Matlab LMI toolbox. A numerical example that confirms the theoretical result is also presented.

Keywords: neural network, LMI, Lyapunov-krasovskii functional.

1 Introduction

In the past few decades, neural network have been studied and developed extensively. In many areas such as combinatorial optimization, signal processing, pattern recognition and many other fields neural networks have been successfully applied. However, successful applications are greatly dependent on the dynamic behavior of neural network, we know that stability is one of the main properties of neural network. So study on the stability of neural network is important[1-4].

In real nervous system, the synaptic transmission is a noisy process brought on by random fluctuation from the release of neurotransmitters and other probabilistic causes. It has also been know that a neural network could be stabilized or destabilized by certain stochastic inputs[5]. Hence, the stability analysis problem for stochastic neural networks becomes increasingly significant. [6] is the original work on the stochastic neural networks and some algebraic criterion of almost sure exponential stability and instability are obtained. Some results related to this problem have been got[7-11]. On the other hand, the connection weights of the neurons depend on certain resistance and capacitance values that include uncertainties. When modeling neural networks, the parameter uncertain should be taken into account, and therefore the problem of robust stability analysis for neural networks becomes very important[12-15]. It should be pointed out that, the robust stability analysis problem for DNSNNs(neural stochastic neural networks with delay)has not been investigated, and remains important and challenging.

This paper studies the global robust stability of stochastic Hopfield neural networks with varying delay, based on Lyapunov-krasovskii functional approach

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 474–479, 2012.

and linear matrix inequality technique. A novel result expressed in terms of linear matrix inequality is proposed, which have the advantage of considering the difference between the neuronal excitatory and the inhibitory effects, and so are less conservative than some earlier ones. Moreover, the proposed result is easy to check and apply, because it can be effectively solved by Matlab LMI toolbox.

2 Model Description and Preliminaries

We consider a delayed stochastic Hopfield neural network described by the following:

$$\dot{x}(t) = [-Dx(t)+Af(x(t))+Bf(x(t-\tau(t)))]dt+\sigma(t,x(t),x(t-\tau(t)))dW(t) \quad (1)$$

where $x(t) = (x_1(t), x_2(t), ..., x_n(t))^T$, are the state vectors at time $t, D = diag\{d_1, d_2, ..., d_n\}$, is self feedback, $d_i \geq 0, i = 1, 2, ...n$. $f(x(t)) = (f_1(x_1(t)), f_2(x_2(t)), ..., f_n(x_n(t)))^T$, are the vectors of outputs, $f_j(x_j(t)), j = 1, 2, ...n$, is activation function of the j-neuron. $A = (a_{ij})_{n\times n}$ is state feedback matrix, $B = (b_{ij})_{n\times n}$ is state delay feedback matrix. $\tau(t) = (\tau_1(t), \tau_2(t), ..., \tau_n(t))^T$, is transmission delay. It meets the condition: $\tau_i \leq \tau, \tau_i'(t) \leq \mu < 1$, $i = 1, 2, ..., n$. $\sigma(t, x(t), x(t - \tau(t)))dW(t)$ is random disturbance, $W(t) = (W_1(t), W_2(t), ..., W_n(t))$ is an m-dimensional Brownian motion defined on a complete probability space (Ω, F, P) with a filtration $\{F_t\}_{t>0}, \sigma : R_+ \times R^n \times R^n \to R^{n\times n}$, meets the local Lipschitz continuous and linear growth condition. The activation function and $\sigma(t, x, x(t - \tau(t)))$ satisfy the following assumptions:

$(H1)$: There exist positive numbers L_j such that $0 \leq \frac{f_j(x_j(t))}{x_j} \leq L_j, f_j(0) = 0, j = 1, 2, ..., n$.
$(H2)$: There are real matrices $C_1 \geq 0, C_2 \geq 0$, and $P > 0$ such that

$$\begin{aligned} &trace[\sigma^T(t, x(t), x(t-\tau(t)))P\sigma(t, x(t), x(t-\tau(t)))] \\ &\leq x^T(t)C_1x(t) + x^T(t-\tau(t))C_2x(t-\tau(t)). \end{aligned}$$

The quantities D, A, B may be intervalized as follows:

$$\begin{aligned} D_I &= \{D = diag(d_i), \underline{D} \leq D \leq \overline{D}, i.e, \underline{d_i} \leq d_i \leq \overline{d_i}, i = 1, 2, ..., n\} \\ A_I &= \{A = (a_{ij})_{n\times n}, \underline{A} \leq A \leq \overline{A}, i.e, \underline{a_{ij}} \leq a_{ij} \leq \overline{a_{ij}}, i = 1, 2, ..., n\} \\ B_I &= \{B = (b_{ij})_{n\times n}, \underline{B} \leq B \leq \overline{B}, i.e, \overline{\underline{b_{ij}}} \leq b_{ij} \leq \overline{b_{ij}}, i = 1, 2, ..., n\} \end{aligned} \quad (2)$$

Definition: The system (1) with the parameter ranges defined by (2) is globally robust stable if the system is globally asymptotically stable for all $D \in D_I, A \in A_I, B \in B_I$.

Lemma 1[15]: For any $x = [x_1, x_2, ..., x_n]^T, y = [y_1, y_2, ..., y_n]^T, A = (a_{ij})_{n\times n}$, $B = (b_{ij})_{n\times n}$ with $|a_{ij}| \leq b_{ij}$, we have:

$$x^TAy \leq |x|^TB|y| \quad (3)$$

3 Main Result

Theorem 1: Under the assumptions (H_1) and (H_2), the trivial solution of system (1) is globally robustly stable if there exist positive diagonal matrix P, Q, Q_1, S, positive definite matrix Q_2, and scalar $\mu > 0$, satisfying the following LMIs:

$$\Pi_1 = \begin{bmatrix} \Lambda & 0 & A^{*T}P & B^{*T}P \\ 0 & -(1-\mu)Q_1 + C_2 & 0 & 0 \\ PA^* & 0 & Q_2 - S & 0 \\ PB^* & 0 & 0 & -(1-\mu)Q_2 \end{bmatrix} < 0 \tag{4}$$

$$\Pi_2 = \begin{bmatrix} -Q + Q_1 + L^TSL & 0 & A_*^TP & B_*^TP \\ 0 & -(1-\mu)Q_1 + C_2 & 0 & 0 \\ PA_* & 0 & Q_2 - S & 0 \\ PB_* & 0 & 0 & -(1-\mu)Q_2 \end{bmatrix} < 0 \tag{5}$$

where $A^* = \frac{1}{2}(\overline{A} + \underline{A})$,$B^* = \frac{1}{2}(\overline{B} + \underline{B})$,$A_* = \frac{1}{2}(\overline{A} - \underline{A})$,$B_* = \frac{1}{2}(\overline{B} - \underline{B})$
$\Lambda = -2P\underline{D} + Q + Q_1 + C_1 + L^TSL.$

Proof: Consider the following Lyapunov functional:

$$\begin{aligned} V(x,t) = x^T(t)Px(t) + 2\int_{t-\tau(t)}^{t} x^T(s)Q_1x(s)ds \\ +2\int_{t-\tau(t)}^{t} f^T(x(s))Q_2f(x(s))ds \end{aligned} \tag{6}$$

Applying It$\hat{o}$'s formula to $V(x,t)$, we get

$$\begin{aligned} \mathcal{L}V(x,t) = & -2x^T(t)PDx(t) + 2x^T(t)PAf(x(t)) + 2x^T(t)PBf(x(t-\tau(t))) \\ & +2x^T(t)Q_1x(t) \\ & -2(1-\tau'(t))x^T(t-\tau(t))Q_1x(t-\tau(t)) \\ & +2f^T(x(t))Q_2f(x(t)) \\ & -2(1-\tau'(t))f^T(x(t-\tau(t)))Q_2f(x(t-\tau(t))) \\ & +\sigma^T(t,x(t),x(t-\tau(t)))P\sigma(t,x(t),x(t-\tau(t))) \end{aligned} \tag{7}$$

From:$\tau_i \leq \tau, \tau_i'(t) \leq \mu < 1, i = 1, 2, ..., n$, we have

$$\begin{aligned} \mathcal{L}V(x,t) \leq & -2x(t)PDx^T(t) \\ & +2x^T(t)PAf(x(t)) \\ & +2x^T(t)PBf(x(t-\tau(t))) \\ & +2x^T(t)Q_1x(t) \\ & -2(1-\mu)x^T(t-\tau(t))Q_1x(t-\tau(t)) \\ & +2f^T(x(t))Q_2f(x(t)) \\ & -2(1-\mu)f^T(x(t-\tau(t)))Q_2f(x(t-\tau(t))) \\ & +x^T(t)C_1x(t) \\ & +x^T(t-\tau(t))C_2x(t-\tau(t)) \end{aligned} \tag{8}$$

Since $p_i, d_i, i = 1, 2, ...n$ are positive constants, we can get that:

$$-2x_i(t)p_id_ix_i \leq -2x_i(t)p_i\underline{d_i}x_i, i = 1, 2, ...n \tag{9}$$

i.e

$$-2x^T(t)PDx(t) \le -2x^T(t)P\underline{D}x(t) \tag{10}$$

According to (H1): $f(x(t)) \le Lx(t)$ there is a matrix S, such that:

$$2x^T(t)L^TSLx(t) \ge 2f^T(x(t))Sf(x(t)) \tag{11}$$

Then rewrite $2x^T(t)PAf(x(t))$ as:

$$2x^T(t)PAf(x(t)) = 2x^T(t)PA^*f(x(t)) + 2x^T(t)PA_0f(x(t)) \tag{12}$$

where $A = A^* + A_0$, according to (4). we get $|a_{0ij}| \le a_{*ij}$, since $p_i \ge 0, i = 1, 2, ..., n$, we get that $|p_i a_{0ij}| \le p_i a_{*ij}$, from lemma 1, we can get that:

$$2x^T(t)PA_0f(x(t)) \le 2|x(t)|^TPA_*|f(x(t))| \tag{13}$$

substituting inequality (13) into (12), we get

$$2x^T(t)PAf(x(t)) \le 2x^T(t)PA^*f(x(t)) + 2|x(t)|^TPA_*|f(x(t))| \tag{14}$$

Similarly:

$$\begin{aligned} &2x^T(t)PBf(x(t-\tau(t))) \le \\ &\qquad 2x^T(t)PB^*f(x(t-\tau(t))) + 2|x(t)|^TPB_*|f(x(t-\tau(t)))| \end{aligned} \tag{15}$$

It is easy to see that:

$$\begin{aligned} &x^T(t)Q_1x(t) = |x(t)|^TQ_1|x(t)| \\ &f^T(x(t))Q_2f(x(t)) = |f(x(t))|^TQ_2|f(x(t))| \\ &x^TQx(t) = |x(t)|^TQ|x(t)| \\ &(1-\mu)x^T(t-\tau(t))Q_1x(t-\tau(t)) = (1-\mu)|x(t-\tau(t))|^TQ_1|x(t-\tau(t))| \\ &(1-\mu)f^T(x(t-\tau(t)))Q_2f(x(t-\tau(t))) \\ &\qquad = (1-\mu)|f(x(t-\tau(t)))|^TQ_2|f(x(t-\tau(t)))| \end{aligned} \tag{16}$$

Using(H2), (8)-(16), we can get that:

$$\begin{aligned} \mathcal{L}V(x,t) \le\ & x^T(-2P\underline{D} + Q)x(t) + 2x^T(t)PA^*f(x(t)) \\ & + 2|x(t)|^TPA_*|f(x(t))| + 2x^TPB^*f(x(t-\tau(t))) \\ & + 2|x(t)|^TPB_*|f(x(t-\tau(t)))| + x^T(t)Q_1x(t) \\ & - (1-\mu)x^T(t-\tau(t))Q_1x(t-\tau(t)) \\ & + |x(t)|^TQ_1|x(t)| \\ & - (1-\mu)|x(t-\tau(t))|^TQ_1|x(t-\tau(t))| \\ & + f^T(x(t))Q_2f(x(t)) \\ & - (1-\mu)f^T(x(t-\tau(t)))Q_2f(x(t-\tau(t))) \\ & + |f(x(t))|^TQ_2|f(x(t))| \\ & - (1-\mu)|f(x(t-\tau(t)))|^TQ_2|f(x(t-\tau(t)))| \\ & + x^T(t)C_1x(t) + x^T(t-\tau(t))C_2x(t-\tau(t)) \\ & - |x(t)|^TQ|x(t)| + x^T(t)L^TSLx(t) - f^T(x(t))Sf(x(t)) \\ & + |x(t)|^TL^TSL|x(t)| - |f(x(t))|^TS|f(x(t))| \\ & = \theta^T(t)\mathit{\Pi}_1\theta(t) + |\theta(t)|^T\mathit{\Pi}_2|\theta(t)| < 0 \end{aligned} \tag{17}$$

where $\theta(t) = [x^T(t), x^T(t-\tau(t)), f^T(x(t)), f^T(x(t-\tau(t)))]$
So, system (1) is globally robustly stable.

4 Example

In this section, we give an example to illustrate the effectiveness of our result.

Considering a delayed stochastic neural network with the following parameters:

$$\underline{D} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \ \overline{D} = \begin{bmatrix} 1.02 & 0 \\ 0 & 1.01 \end{bmatrix} \ \underline{A} = \begin{bmatrix} 0.01 & -0.035 \\ 0.03 & 0.01 \end{bmatrix} \ \overline{A} = \begin{bmatrix} 0.03 & -0.025 \\ 0.03 & 0.03 \end{bmatrix}$$

$$\underline{B} = \begin{bmatrix} -0.025 & -0.015 \\ -0.016 & -0.025 \end{bmatrix} \ \overline{B} = \begin{bmatrix} 0.075 & 0.085 \\ -0.014 & 0.075 \end{bmatrix}$$

$f_j(x_j) = tanh(x_j), j = 1, 2.$

So, we can get that:

$$A^* = \begin{bmatrix} 0.02 & -0.03 \\ 0.06 & 0.02 \end{bmatrix} \ A_* = \begin{bmatrix} 0.01 & 0.005 \\ 0 & 0.01 \end{bmatrix} \ B^* = \begin{bmatrix} 0.025 & 0.035 \\ -0.015 & 0.025 \end{bmatrix}$$

$$B_* = \begin{bmatrix} 0.05 & 0.05 \\ 0.001 & 0.05 \end{bmatrix} \ L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$C_1 = 0.06, C_2 = 0.06, \mu = 0.3.$

By the LMI toolbox in Matlab, we can get a feasible solution of LMIs (4) (5):

$$P = \begin{bmatrix} 65.2089 & 0 \\ 0 & 65.3225 \end{bmatrix} \ Q = \begin{bmatrix} 61.7629 & 0 \\ 0 & 62.5716 \end{bmatrix} \ Q_1 = \begin{bmatrix} 18.6049 & 0 \\ 0 & 18.6580 \end{bmatrix}$$

$$Q_2 = \begin{bmatrix} 17.4050 & -0.1109 \\ -0.1109 & 17.1270 \end{bmatrix} \ S = \begin{bmatrix} 30.6516 & 0 \\ 0 & 30.7814 \end{bmatrix}$$

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grant No. 10902065.

References

1. Singh, V.: A generalized LMI-based approach to the global asymptotic stability of delayed cellular neural networks. IEEE Trans. Neural Network 15, 223–225 (2004)
2. Arik, S.: An analysis of exponential atability of delay neural networks with time varing delays. Neural Network 17, 1027–1031 (2004)
3. Cao, J., Yuan, K., Li, H.: Global asymptotical stability of recurrent neural networks with multiple discrete delays and distributed delays. IEEE Trans. Neural Network 17, 1646–1651 (2006)
4. Liu, X., Jiang, N.: Robust stability analysis of generalized neural networks with multiple discrete delays and multiple distributed delays. Neurocomuting 72, 1789–1796 (2009)
5. Blythe, S., Mao, X., Liao, X.: Stability of stochastic delay neural networks. Journal of the Franklin Institute 338, 481–495 (2001)
6. Liao, X., Mao, X.: Stability of stochastic neural networks. Neural, Parallel Scientific Computations 4, 205–224 (1996)
7. Liao, X., Mao, X.: Exponential stability and instability of stochastic neural networks. Stochastic Analysis and Applications 14, 165–185 (1996)

8. Huang, H., Ho, D., Lam, J.: Stochastic stability analysis of fuzzy Hopfield neural networks with time-varying delays. IEEE Trans. Circuits Syst. 52, 251–255 (2005)
9. Lou, X., Cui, B.: Delay-dependent stochastic stability of delayed Hopfield neural networks with Markovian jump parameter. Journal of Mathematical Analysis and Applications 328, 316–326 (2007)
10. Wang, Z.: Robust stability for stochastic Hopfield neural networks with time delay. Nonlinear Analysis, Real-world Application 7, 1119–1128 (2006)
11. Lou, X., Cui, B.: Stochastic Robust Stability of Markovian Jump Nonlinear Uncertain Neural Networks with Wiener Process. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3971, pp. 165–171. Springer, Heidelberg (2006)
12. Cao, J., Huang, D., Qu, Y.: Global robust stability of delayed recurrent neural networks. Chaos, Solitons Franctals 123, 221–229 (2005)
13. Cao, J., Chen, T.: Globally exponentially robust stability and periodicity of delay neural networks. Chaos, Solitons Fractals 22, 957–963 (2004)
14. Xu, S., Lam, J., Ho, D.: Novel global robust stability criteria for interval neural networks with multiple time-varying delays. Phys. Lett. A 342, 322–331 (2005)
15. Shen, T., Zhang, Y.: Improve global robust stability criteria for delayed neural networks. IEEE, Trans. Circuits Syst. 54, 715–719 (2007)

RST-DCA: A Dendritic Cell Algorithm Based on Rough Set Theory

Zeineb Chelly and Zied Elouedi

LARODEC, University of Tunis, 2000 Le Bardo, Tunisia
zeinebchelly@yahoo.fr, zied.elouedi@gmx.fr

Abstract. The Dendritic Cell Algorithm (DCA) is an immune-inspired classification algorithm based on the behavior of dendritic cells. The DCA performance depends on its data pre-processing phase including feature selection and their categorization to specific signal types. For feature selection, DCA applies the principal component analysis (PCA). Nevertheless, PCA does not guarantee that the selected first principal components will be the most adequate for classification. Furthermore, the categorization of features to their specific signal types is based on the PCA attributes' ranking in terms on variability which does not make "sense". Thus, the aim of this paper is to develop a new DCA data pre-processing method based on Rough Set Theory (RST). In this newly-proposed hybrid DCA model, the selection and the categorization of attributes are based on the RST CORE and REDUCT concepts. Results show that using RST instead of PCA for the DCA data pre-processing phase yields much better performance in terms of classification accuracy.

Keywords: Artificial immune systems, Dendritic Cells, Rough Sets, Core, Reduct.

1 Introduction

Artificial Immune Systems (AIS) are a class of computationally intelligent systems inspired by the principles of the vertebrate immune system. As AIS is being developed significantly, novel algorithms termed "2nd Generation AISs" have been created. One such 2nd Generation AIS is the Dendritic Cell Algorithm (DCA) [5] which is based on the behavior of the natural "dendritic cells" (DCs). DCA has been successfully applied to various applications. In fact, its performance depends on its data pre-processing phase which is divided into two main steps: feature selection and signal categorization. More precisely, DCA uses the principal component analysis (PCA) to automatically select features and to categorize them to their specific signal types; as danger signals (DS), as safe signals (SS) or as pathogen-associated molecular patterns (PAMP)[6]. DCA combines these signals with location markers in the form of antigen to process his classification task. For signal selection, PCA transforms a finite number of possibly correlated vectors into a smaller number of uncorrelated vectors, termed "principal components" which reveals the internal structure of the given data with the

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 480–487, 2012.

focus on data variance [6]. However, using PCA for feature selection presents a drawback as it is not necessarily true that the first selected components will be the adequate features to retain [7]. Thus, the choice of these components for the DCA can influence its classification task by producing unreliable results. As for feature categorization, DCA uses the generated PCA ordered list of standard deviation values to assign for each selected attribute its signal type (SS, DS or PAMP). However, this categorization process which is based on high and low values of the calculated standard deviations does not make “sense” as a coherent process which can influence negatively the DCA functioning. Thus, in this paper, we develop a novel AIS hybrid model based on a new automatic data pre-processing phase for the DCA. As DCA was hybridized with various techniques to improve its classification performance such as with fuzzy set theory [2], a fuzzy clustering technique [3] and a maintenance policy [4], in this paper, our new hybrid model named “RST-DCA” is grounded on the behavior of DCs within the framework of Rough Set Theory (RST). Our RST-DCA model uses the RST REDUCT and CORE concepts to select the right features to retain and to categorize them into their right signal types. This paper is structured as follows: Section 2 of this paper introduces the DCA. Section 3 presents the RST concepts. Section 4 details our hybrid RST-DCA AIS system. The experiments and the results are outlined in Section 5 and 6.

2 The Dendritic Cell Algorithm

The first DCA step is data pre-processing which includes feature selection and signal categorization. For signal selection, DCA applies the PCA that reduces data dimension, by accumulating the vectors that can be linearly represented by each other [6]. Once features are selected, PCA is applied to assign each attribute to its specific signal type. More precisely, DCA uses the PCA calculated standard deviations and selects the highest values. As both PAMP and SS are positive indicators of an anomalous and normal signal [5], one attribute is used to form both PAMP and SS. Thus, the attribute having the lowest standard deviation out of the selected attribute set is used to form both PAMP and SS. Using one attribute for these two signals requires a threshold level to be set: values greater than this can be classed as SS otherwise as PAMP [5]. As for the DS attribute assignment and since the DS is “less than certain to be anomalous”, the combination of the rest of the selected attributes are chosen to represent it [5]. After calculating the values of SS, PAMP and DS [5], DCA adheres these signals and antigen to fix the context of each DC. DCA processes its input signals to decide whether the collected DC goes to the semi-mature context, implying that the antigen data is normal, or if the DC goes to the mature context, signifying an anomalous data item. The nature of the response is determined by measuring the number of fully mature DCs and is represented by the Mature Context Antigen Value (MCAV). $MCAV$ is used to assess the degree of anomaly of a given antigen. By applying thresholds at various levels, analysis can be performed to assess the anomaly detection capabilities of the algorithm. Those antigens whose

$MCAV$ are greater than the anomalous threshold are classified as anomalous else as normal. More DCA details and its pseudocode can be found in [5].

3 Rough Set Theory

In RST [8], an *information table* is defined as a tuple $T = (U, A)$ where U and A are two finite, non-empty sets, U the *universe* of primitive objects and A the set of attributes. A may be partitioned into C and D, called *condition* and *decision* attributes, respectively. Let $P \subset A$ be a subset of attributes. The indiscernibility relation, $IND(P)$, is an equivalence relation defined as: $IND(P) = \{(x, y) \in U \times U : \forall a \in P, a(x) = a(y)\}$, where $a(x)$ denotes the value of feature a of object x. The family of all equivalence classes of $IND(P)$ is denoted by $U/IND(P)$. Equivalence classes $U/IND(C)$ and $U/IND(D)$ are respectively called *condition* and *decision* classes. For any concept $X \subseteq U$ and attribute subset $R \subseteq A$, X could be approximated by the R-*lower* and R-*upper* approximations using the knowledge of R. The X lower approximation is the set of objects U that are surely in X, defined as: $\underline{R}(X) = \bigcup\{E \in U/IND(R) : E \subseteq X\}$. The X upper approximation is the set of U objects that are possibly in X, defined as: $\overline{R}(X) = \bigcup\{E \in U/IND(R) : E \cap X \neq \emptyset\}$. The boundary region is defined as: $BND_R(X) = \overline{R}(X) - \underline{R}(X)$. If $BND_R(X)$ is empty, $\overline{R}(X) = \underline{R}(X)$, X is said to be R-*definable*. Otherwise X is a rough set with respect to R. The positive region of $U/IND(D)$ with respect to C is denoted by $POS_c(D)$ where: $POS_c(D) = \bigcup \overline{R}(X)$. $POS_c(D)$ is a set of objects of U that can be classified with certainty to classes $U/IND(D)$ employing attributes of C. For feature selection, RST defines two main concepts; the CORE and the REDUCT. The CORE is equivalent to the set of strong relevant features which are *indispensable* attributes in the sense that they cannot be removed without loss of prediction accuracy of the original database. The REDUCT is a combination of all strong relevant features and some weak relevant features that can sometimes contribute to prediction accuracy. These concepts provide a good foundation upon which we can define our basics for defining the importance of each attribute. In RST, a subset $R \subseteq C$ is said to be a D-*reduct* of C if $POS_R(D) = POS_C(D)$ and there is no $R' \subset R$ such that $POS_{R'}(D) = POS_C(D)$. In other words, the REDUCT is the minimal set of attributes preserving the positive region. There may exist many reducts (a family of reducts), $RED^F_D(C)$, in T. The CORE is the set of attributes that are contained by all reducts, defined as: $CORE_D(C) = \bigcap RED_D(C)$ where $RED_D(C)$ is the D-reduct of C. In other words, the CORE is the set of attributes that cannot be removed without changing the positive region. This means that all attributes present in the CORE are indispensable.

4 RST-DCA: The Solution Approach

4.1 RST-DCA Feature Selection Process

Our learning problem is to select high discriminating features for antigen classification from the original input data set which corresponds to the antigen information database. We may formalize this problem as an information table, where universe $U = \{x_1, x_2, \ldots, x_N\}$ is a set of antigen identifiers, the conditional attribute set $C = \{c_1, c_2, \ldots, c_N\}$ contains each feature of the information table to select and the decision attribute D of our learning problem corresponds to the class label of each sample. As DCA is applied to binary classification problems, the input database has a single binary decision attribute. Hence, the decision attribute D, which corresponds to the class label, has binary values d: either the antigen is collected under safe circumstances reflecting a normal behavior (classified as normal) or the antigen is collected under dangerous circumstances reflecting an anomalous behavior (classified as anomalous). The condition attribute feature D is defined as follows: $D = \{normal, anomalous\}$. For that, RST-DCA computes, first of all, the positive region for the whole attribute set C for both label classes of D: $POS_C(\{d\})$. Based on the RST computations (seen previously in Section 3), RST-DCA computes the positive region of each feature c and the positive region of all the composed features $C - \{c\}$ (when discarding each time one feature c from C) defined respectively as $POS_c(\{d\})$ and $POS_{C-\{c\}}(\{d\})$, until finding the minimal subset of attributes R from C that preserves the positive region as the whole attribute set C does. In fact, RST-DCA removes in each computation level the unnecessary features that may affect negatively the accuracy of the RST-DCA. The result of these computations is either one reduct $R = RED_D(C)$ or a family of reducts $RED^F_D(C)$. Any reduct of $RED^F_D(C)$ can be used to replace the original antigen information table. Consequently, if the RST-DCA generates only one reduct $R = RED_D(C)$ then for the feature selection process, RST-DCA chooses this specific R which represents the most informative features that preserve nearly the same classification power of the original data set. If the RST-DCA generates a family of reducts $RED^F_D(C)$ then RST-DCA chooses randomly one reduct R among $RED^F_D(C)$ to represent the original input antigen information table. This random choice is argued by the same priority of all the reducts in $RED^F_D(C)$. In other words, any reduct R of the reducts $RED^F_D(C)$ can be used to replace the original information table. These attributes which constitute the reduct will describe all concepts in the original training data set. By using the REDUCT, our method can guarantee that the selected attributes will be the most relevant for its classification task.

4.2 RST-DCA Feature Categorization Process

RST-DCA has to assign, now, for each selected attribute, produced by the previous step, its specific signal type; either as PAMP, as DS or SS. As previously stated, both PAMP and SS have a certain final context (either an anomalous or a normal behavior) while the DS cannot specify exactly the final context to assign to the collected antigen as the DS may or may not indicate an anomalous situation. This problem can be formulated as follows: Both PAMP and SS are more informative than DS which means that both of these signals can be seen as indispensable attributes. To define this level of importance, our method uses the CORE RST concept. As for DS, it is less informative than PAMP and SS. Therefore, RST-DCA uses the rest of the REDUCT attributes (discarding the attributes of the CORE chosen to represent both SS and PAMP) to represent the DS. As stated in the previous step, our method may either produce only one reduct R or a family of reducts $RED_D^F(C)$. The process of signal categorization for both cases are described in what follows: In case where our RST-DCA generates only one reduct; it means that $CORE_D(C) = RED_D(C)$. In other words, all the features of the reduct are indispensable. In this case, RST-DCA selects randomly one attribute c from $CORE_D(C)$ and assigns it to both PAMP and SS as they are the most informative signals. Using one attribute for these two signals requires a threshold level to be set: values greater than this can be classed as SS, otherwise as a PAMP signal. The rest of the attributes $CORE_D(C) - \{c\}$ are combined and the resulting value is assigned to the DS as it is less than certain to be anomalous. In case where our RST-DCA produces a family of reducts $RED_D^F(C)$, the RST-DCA presents both concepts: the core $CORE_D(C)$ and the reduct $RED_D^F(C)$. Let us remind that $CORE_D(C) = \bigcap RED_D(C)$; which means that on one hand we have the minimal set of attributes preserving the positive region (reducts) and on the other hand we have the set of attributes that are contained in all reducts (core) which cannot be removed without changing the positive region. This means that all the attributes present in the CORE are indispensable. For signal categorization, PAMP and SS are assigned, randomly, one attribute c among the features in $CORE_D(C)$. As for the DS signal assignment, RST-DCA chooses, randomly, a reduct $RED_D(C)$ among $RED_D^F(C)$. Then, RST-DCA combines all the $RED_D(C)$ features except that c attribute already chosen and assigns the resulting value to the DS. Once signal categorization is achieved, RST-DCA processes its next steps as the DCA does [5].

5 Experimental Setup

To test the validity of our RST-DCA hybrid model, our experiments are performed using binary databases from [1] described in Table 1.

For data pre-processing, DCA and RST-DCA uses PCA and RST, respectively. Each data item is mapped as an antigen, with the value of the antigen equal to the data ID of the item. To perform anomaly detection, a threshold which is automatically generated from the data is applied to the MCAVs. The MCAV threshold is derived from the proportion of anomalous data instances of

Table 1. Description of Databases

Database	Ref	♯ Instances	♯ Attributes
Spambase	SP	4601	58
SPECTF Heart	SPECTF	267	45
Cylinder Bands	CylB	540	40
Chess	Ch	3196	37
Ionosphere	IONO	351	35
Mushroom	Mash	8124	23
Congressional Voting Records	CVT	435	17
Tic-Tac-Toe Endgame	TicTac	958	10

the whole data set. Items below the threshold are classified as class 1 and above as class 2. The resulting classified antigens are compared to the labels given in the original data sets. The results presented are based on mean MCAV values generated across 10 runs. We evaluate the performance of RST-DCA in terms of number of extracted features, sensitivity, specificity and accuracy which are defined as: $Sensitivity = TP/(TP + FN)$; $Specificity = TN/(TN + FP)$; $Accuracy = (TP + TN)/(TP + TN + FN + FP)$; where TP, FP, TN, and FN refer respectively to: true positive, false positive, true negative and false negative. We will also compare the classification performance of our RST-DCA method to well known classifiers which are the Support Vector Machine (SVM), Artificial Neural Network (ANN) and to the Decision Tree (DT).

6 Results and Discussion

In this Section, we show that using RST instead of PCA is much convenient for the DCA data pre-processing phase as it improves its classification performance which is confirmed by the results given in Table 2. Let us remind that for signal selection, DCA applies PCA where it selects the highest standard deviation values. As the highest values have to be selected, this needs either to keep only the eigenvalues larger than 1 [7] or involving the user to decide which features to keep for the algorithm. However, the fact of using eigenvalues can either lead to overestimate the number of factors to keep or to underestimate it leading to ignore important information. In addition, involving users to determine a priori the number of attributes to retain may result to preserve more or less features than necessary. In this Section, we will show that these problems are solved by our RST-DCA.

From Table 2, it is clearly seen that the number of features selected by our RST-DCA is less than the one generated by DCA when applying PCA (PCA-DCA). This can be explained by the appropriate use of RST for feature selection. In fact, RST-DCA keeps only the most informative features which constitute the REDUCT. For instance, by applying our RST-DCA method to the CylB data set, the number of selected features is only 7 attributes. However, when applying

Table 2. DCA and RST-DCA Comparison Results

Database	Sensitivity (%)		Specificity (%)		Accuracy (%)		♯ Attributes	
	DCA		DCA		DCA		DCA	
	PCA	RST	PCA	RST	PCA	RST	PCA	RST
SP	86.76	94.53	87.58	94.47	87.26	94.5	14	8
SPECTF	72.16	84.43	67.27	74.54	71.16	82.4	11	4
CylB	91.50	96.50	92.94	96.79	92.38	96.67	16	7
Ch	94.06	97.84	93.64	98.23	93.86	98.02	14	11
IONO	93.65	95.23	94.22	96.88	94.58	96.29	24	19
Mash	99.41	99.82	99.28	99.73	99.34	99.77	7	6
CVT	91.07	95.83	92.13	97	91.72	96.55	14	8
TicTac	91.37	93.45	89.15	93.67	90.6	93.52	7	6

the PCA-DCA to the same database (CylB), the number of the retained features is 16. We can notice that PCA preserves additional features which are the result of the PCA overestimation of the number of factors to retain. This overestimation affects the DCA classification task by producing unreliable results. On the other hand, RST-DCA based on the REDUCT concept, selects the minimal set of features from the original database and can guarantee that the reduct attributes will be the most relevant for its classification task. In fact, by reducing more the number of features while preserving the classification power of the original data set, our RST-DCA has the advantages to decrease the cost of acquiring data and to make the classification model easier to understand unlike when applying the PCA. In addition, RST-DCA has sufficient advantages over the PCA-DCA, as it does not require any additional information about data a priori such as thresholds or expert knowledge on a particular domain. Thus, RST-DCA results will not be influenced by any external information. As for the classification accuracy, from Table 2, we can easily remark that the RST-DCA accuracy is notably better than the one given by the PCA-DCA. For example, when applying the RST-DCA to the CylB database, the RST-DCA accuracy is set to 96.67%. Nevertheless, when applying the PCA-DCA to the same database, the accuracy is 92.38%. Same remark is noticed for both the sensitivity and the specificity criteria. These encouraging RST-DCA results are explained by the appropriate set of features selected and their categorization to their right and specific signal types. As stated previously, the classification results of the DCA depends on its data pre-processing phase which is crucial to obtain reliable results. RST-DCA uses the REDUCT RST fundamental concept to select only the essential part of the original database. This pertinent set of minimal features can guarantee a solid base for the signal categorization step. The RST-DCA good classification results are also explained by the appropriate categorization of each selected signal to its right signal type by using both the REDUCT and the CORE concepts. As for DCA, by applying the PCA, it produces less accuracy in comparison to our RST-DCA method which is explained by the inappropriate use of the PCA for data pre-processing. In fact, the first components selected are not necessarily the right set of features to retain since this set still contains extra features that do not add anything new to the target concept while increasing the cost of acquiring data. The set may also contain misleading features which have a negative effect on classification accuracy. Furthermore, the DCA categorization step does not make "sense" as a coherent categorization procedure.

The performance of our RST-DCA is, also, compared to SVM, ANN and to DT in terms of the average of accuracies on the 8 data sets. The parameters of SVM, ANN and DT are set to the most adequate parameters to these algorithms using the Weka software. Figure 1 shows that PCA-DCA has nearly the same classification performance as SVM and ANN and a better one than DT. It also shows that our RST-DCA outperforms all the mentioned classifiers including the PCA-DCA in terms of overall accuracy. These encouraging RST-DCA results are explained by the appropriate application of RST to the DCA data pre-processing phase making the DCA a better classifier by generating pertinent and more reliable results.

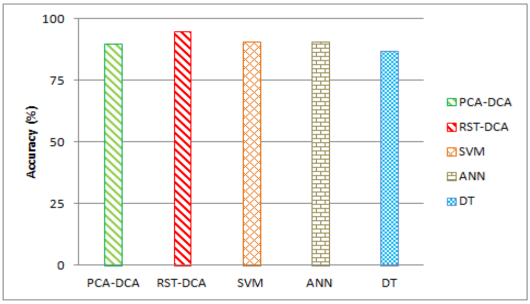


Fig. 1. Comparison of Classifiers' Average Accuracies on the 8 Binary Datasets

7 Conclusion and Further Works

In this paper, we have introduced a new hybrid computational biological model for the DCA based on RST. Our model aims to select the convenient set of features from the initial database and to perform their signal categorization using the REDUCT and the CORE RST concepts. The experimentation results show that our RST-DCA is capable of performing better its classification task than DCA and other classifiers. Future works will include the use of fuzzy rough set theory for the DCA and the application of RST-DCA to real world problems.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2007)
2. Chelly, Z., Elouedi, Z.: FDCM: A Fuzzy Dendritic Cell Method. In: Hart, E., McEwan, C., Timmis, J., Hone, A. (eds.) ICARIS 2010. LNCS, vol. 6209, pp. 102–115. Springer, Heidelberg (2010)
3. Chelly, Z., Elouedi, Z.: Further Exploration of the Fuzzy Dendritic Cell Method. In: Liò, P., Nicosia, G., Stibor, T. (eds.) ICARIS 2011. LNCS, vol. 6825, pp. 419–432. Springer, Heidelberg (2011)
4. Chelly, Z., Smiti, A., Elouedi, Z.: COID-FDCM: The Fuzzy Maintained Dendritic Cell Classification Method. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 233–241. Springer, Heidelberg (2012)
5. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 153–167. Springer, Heidelberg (2005)
6. Gu, F., Greensmith, J., Oates, R., Aickelin, U.: Pca 4 dca: The application of principal component analysis to the dendritic cell algorithm. CoRR (2010)
7. Kaiser, H.: A note on guttmans lower bound for the number of common factors. British Journal of Mathematical and Statistical Psychology 14, 1–2 (1961)
8. Pawlak, Z., Rough, S.: International Journal of Computer and Information Science 11, 341–356 (1982)

A Meta-Learning Approach to Select Meta-Heuristics for the Traveling Salesman Problem Using MLP-Based Label Ranking

Jorge Kanda[1,2], Carlos Soares[3], Eduardo Hruschka[1], and Andre de Carvalho[1]

[1] Instituto de Ciencias Matematicas e de Computacao, Universidade de Sao Paulo,
Avenida Trabalhador Sao-Carlense, 400 - Centro, 13566-590 Sao Carlos - SP, Brazil
{kanda,erh,andre}@icmc.usp.br

[2] Instituto de Ciencias Exatas e Tecnologias, Universidade Federal do Amazonas,
Rua Nossa Senhora do Rosario, 3863 - Tiradentes, 69103-128 Itacoatiara - AM, Brazil

[3] INESC TEC Porto LA/Faculdade de Economia, Universidade do Porto,
Rua Dr. Roberto Frias, 4200-464 Porto, Portugal
csoares@fep.up.pt

Abstract. Different meta-heuristics (MHs) may find the best solutions for different traveling salesman problem (TSP) instances. The *a priori* selection of the best MH for a given instance is a difficult task. We address this task by using a meta-learning based approach, which ranks different MHs according to their expected performance. Our approach uses Multilayer Perceptrons (MLPs) for label ranking. It is tested on two different TSP scenarios, namely: *re-visiting customers* and *visiting prospects*. The experimental results show that: 1) MLPs can accurately predict MH rankings for TSP, 2) better TSP solutions can be obtained from a label ranking compared to multilabel classification approach, and 3) it is important to consider different TSP application scenarios when using meta-learning for MH selection.

Keywords: meta-learning, label ranking, multilayer perceptron, traveling salesman problem.

1 Introduction

The Traveling Salesman Problem (TSP) is a classic optimization problem, which is formally defined by means of a weighted graph $G = (V, E)$, in which $V = \{v_1, v_2, ..., v_n\}$ is a set of vertices and $E = \{\langle v_i, v_j \rangle \colon v_i, v_j \in V\}$ is a set of edges. Each vertex $v_i \in V$ represents a city and each edge $\langle v_i, v_j \rangle \in E$ connects the vertices v_i and v_j. The cost of travel from v_i to v_j is given by the weight value of the edge $\langle v_i, v_j \rangle$. The best solution for a TSP instance involves finding the minimal cost tour visiting each of n cities only once and returning to the starting city [1].

It is difficult to find the best solution for several TSP instances, since this problem belongs to the class of problems known as NP-complete [16]. The TSP complexity is factorial with the number of cities, thus exhaustive search methods

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 488–495, 2012.

present a high computational cost even for small TSP instances. For example, there are approximately 1.22×10^{17} feasible solutions for a TSP with 20 cities.

Good solutions for TSP can be quickly found by different meta-heuristics (MHs) — *e.g.*, Genetic Algorithms [13], and Ant Colony [5]. MHs are search methods that try to escape from local optima through of interaction between local improvement procedures and higher level strategies [8]. Each MH has its own bias which makes it more suitable for a particular class of instances [25]. Thus, given the large number of available MHs, there can be a MH that is the best for a new TSP instance.

Recently, a meta-learning approach addressed the problem of recommending MHs for new TSP instances as a multilabel classification task [14]. However, when multiple MHs are recommended, no guidance is provided concerning the order in which they should be executed. In this work, we address this problem by using a label ranking approach [4] to predict a ranking of MHs, according to their expected performance. Additionally, previous approaches do not distinguish between different TSP scenarios. Here we separately investigate two important scenarios: when the salesperson (re-)visits current customers and when the prospects are visited for the first time.

The remainder of this paper is organized as follows. Section 2 provides a brief background on meta-learning for algorithm selection and on label ranking. The adaptation of MLPs to learn label rankings is discussed in Section 3. Practical application scenarios of interest are described in Section 4. Based on such scenarios, the experimental setting is detailed in Section 5, and the results are reported in Section 6. Finally, the conclusions are presented in Section 7.

2 Meta-Learning and Label Ranking

The selection of the best algorithm for a given problem has been dealt with in Machine Learning (ML) with meta-learning [2]. Meta-learning studies how learning systems can increase in efficiency through experience and how learning itself can become flexible according to the domain or task under study [24].

Studies that relate ML and optimization problems are recent [20]. Concerning the TSP, a meta-learning approach to recommend MHs [14] classifies TSP instances according to the solutions obtained by a set of MHs. As the best solution for a given TSP instance may be achieved by more than one MH, multilabel classification techniques are applied. In [19], MLP-based models are induced to predict the search effort that each algorithm will need to find the best solution.

The induction of a meta-learning model to select MHs for the TSP is illustrated in Figure 1. TSP properties (meta-features) are calculated to a set of TSP instances. Each instance corresponds to one meta-example in the meta-data. A meta-example is labeled by the performance of different MHs when applied to TSP instance. The meta-data is used by a ML technique to induce a meta-model.

In this work, meta-learning is addressed as a label ranking task [7]. In label ranking, the learning problem is to map the instances x from a dataset X to rankings $\succ_x$ (total strict orders) over a finite set of labels $\mathcal{L} = \{\lambda_1, ..., \lambda_m\}$,

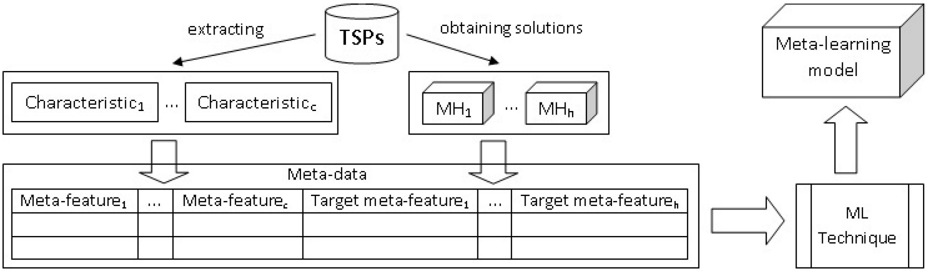


Fig. 1. Meta-learning approach to select meta-heuristics for the TSP

where $\lambda_i \succ_x \lambda_j$ means that, for instance x, label λ_i is preferred to λ_j. A ranking over $\mathcal{L}$ can be represented by a permutation as there exists a unique permutation τ such that $\lambda_i \succ_x \lambda_j$ iff $\tau(\lambda_i) < \tau(\lambda_j)$, where $\tau(\lambda_i)$ denotes the position of the label λ_i in the ranking. A survey on label ranking is presented in [23].

3 Training MLPs for Label Ranking

Since MLPs presented a good performance on a similar problem [19], we use them to rank MHs in this study. In a meta-learning context to rank labels, the input values of the MLP [18] are meta-feature values for a TSP instance. The output layer of the MLP produces a ranking of MHs for this TSP instance. The MH identified in the top position (i.e., rank 1) is the most promising one for this instance.

It is worth noting that the back-propagation algorithm is guided by a regression error measure (e.g., mean squared error) rather than a ranking accuracy measure (e.g., Spearman's correlation coefficient). However, by using a single network to learn the ranks of all labels, the weights to the output layer represent patterns that are specific to the corresponding label. On the other hand, given that there is a single set of weights to the hidden layer, they represent patterns in the data that are common to all the labels and act as latent features.

4 Recommendation Scenarios

Previous approaches [19,14] have considered a single scenario: the recommendation of MHs for instances in which the salesperson revisits current customers. We consider an additional scenario in which the meta-learning approach is used to recommend MHs for instances that contain new customers. To illustrate these scenarios, consider that a company visits clients in different cities and, for simplicity, that there exists only one client in each city. Thus, the recommendation scenarios investigated in our experiments are as follows:

***Revisiting customers* scenario.** The clients in the new instance are a subset of the ones that have been previously visited. Given instances concerning different subsets of the set of cities (e.g., {New York, Washington DC, Boston, Philadelphia}), we would like to know the most promising MH in order to define a route to visit another subset of those cities (e.g., {New York, Boston, Philadelphia}). The intersection between the cities in different instances is non-empty.

Prospect visits **scenario.** All clients on the new route have never been visited in previous routes. Given instances concerning different subsets of the set of cities (e.g., {New York, Washington DC, Boston, Philadelphia}), we would like to know the most promising MH in order to define a route to visit a different set of cities (e.g., {Edinburgh, London, Liverpool, Bristol}). The intersection between the sets of cities in new and old instances is empty.

5 Experimental Setup

A predictive ability of a learning model depends on the significant amount of instances used to train it [22]. We generated several TSP subproblems from benchmark instances extracted from the TSPLIB library [17].

Let $P = \{p_1, ..., p_k\}$ be the set of real TSP instances extracted from the TSPLIB library. A set of subproblems $S_i = \{s_{i,1}, ..., s_{i,z}\}$ can be generated from $p_i \in P$. Thus, the meta-data is a set of meta-examples $X^P = \{x_{1,1}, ..., x_{1,z}, ..., x_{k,1}, ..., x_{k,z}\}$, where each $x_{i,j}$ corresponds to $s_{i,j}$ that represents the j-th subproblem generated from the i-th instance of the real TSP, p_i. For each scenario, the TSP subproblems were generated as follows.

Revisiting customers scenario: 1000 TSP instances were generated from 10 TSP files, $P = \{d1655, fl1400, fnl4461, nrw1379, pcb3038, pr2392, rat783, rl1889, u1817, vm1748\}$. From each $p_i \in P$, 10 subproblems were generated for each of ten different quantities of cities (10, 20, ..., 100), resulting in $S_i = \{s_{i,1}, ..., s_{i,100}\}$.

Prospect visits scenario: 300 TSP instances were generated from 30 TSP files, $P = \{a280, berlin52, bier127, ch130, d1655, d15112, eil101, fl417, fl3795, fnl4461, kroA200, kroB100, kroC100, kroD100, kroE100, linhp318, lin318, nrw1379, p654, pcb3038, pr2392, rat783, rd400, rl1889, rl11849, ts225, tsp225, u1817, usa13509, vm1748\}$. The cities of each $p_i \in P$ were randomly distributed into 10 equal-sized sets. Each set of cities was used to generate a subproblem $s_{i,j}$ that belongs to $S_i = \{s_{i,1}, ..., s_{i,10}\}$.

Five MHs have been used in our experiments: Tabu Search (TS) [9], GRASP (GR) [6], Simulated Annealing (SA) [15], Genetic Algorithms (GA) [13], and Ant Colony (AC) [5]. The following parameter settings were used: TS: tabu list size = 2; number of iterations with no improvement of the current solution = 2; GR: number of iterations = 10; level of randomness and greedy search = 0.5; SA: initial temperature = 1; acceptance rate of neighbor solution = 0.9; cooling rate = 0.01; GA: PMX [10] as the crossover operator; population size = 20; mutation rate = 5%; elitism selection; AC: number of ants = 5; pheromone evaporation rate = 0.5; pheromone influence = 1; heuristic information influence = 1.

These parameter values were chosen after performing some preliminary experiments — just to ensure that every MH could find a reasonable solution for the

TSP instances in hand. Our goal was not to optimize the performance of each MH or promote any particular MH. Instead, we focus on the prediction of the ranking of MHs, with particular emphasis on how the user can take advantage of that ranking to get better solutions for TSP instances.

As these MHs are stochastic, every MH was run 30 times (with same processing time and different initial seeds) for each TSP instance. The average cost of the route of the 30 solutions was used as performance of each MH to compose the ranking of MHs.

Our MLP-based meta-learning models were trained with the standard *back-propagation* algorithm [1] and ten-fold *cross-validation* methodology [12]. We used 14 neurons in the input layer which correspond to 14 meta-features based on the measurements of edges and vertices proposed in [14]. The output layer has five neurons that identify the ranks of the five MHs for the TSP instance provided in the MLP input. The best configuration of the hidden layer is problem-dependent [12]. Therefore, we used the default number of hidden neurons proposed in [11]. For the multilabel classification, we use the binary classification method that has also been successfully applied to classify instances of TSP [14].

6 Experimental Evaluation

We compute the Spearman coefficient (r_S) [21] for every pair of $\langle predicted, ideal\rangle$ (vectors of) ranking and then we average the results. These results are compared to a baseline (the average ranking over the whole dataset [3]). In order to analyze if the performance difference between the proposed approach and baseline is significant, results of the statistical t-test are presented.

Top-N strategy [3] was used to compare the results of our ranking-based approach with the multilabel approach. This strategy evaluates the ranking of MHs by assessing the compromise between the quality of the solutions (cost of the routes) and the cost to obtain them (run time). The quality of the solutions provided by the top-N MHs is given by the best solution among those generated by all MHs. The cost is computed as the sum of the run times of those MHs. As the multilabel classification model does not suggest a ranking of MHs, its average performance is identified by a single point in figures 2a and 2b.

6.1 Experimental Results

Considering r_S as measure to evaluate the predictive models performance, our meta-learning based approach provided good ranking predictions. In particular, average scores $\bar{r_S} = 0.96$ and $\bar{r_S} = 0.93$ were obtained for the scenarios: *revisiting customers* and *prospect visits*, respectively, whereas the baseline model obtained $\bar{r_S} = 0.89$ and $\bar{r_S} = 0.83$, respectively. By applying the t-test to compare the $\bar{r_S}$ values, p-values of 7.15×10^{-42} and 2.61×10^{-12} were obtained for the respective scenarios. These results show that at the 95% confidence level, the performance of the proposed model is significantly better than the baseline model.

[1] Using the default values of the nnet package (R programming language).

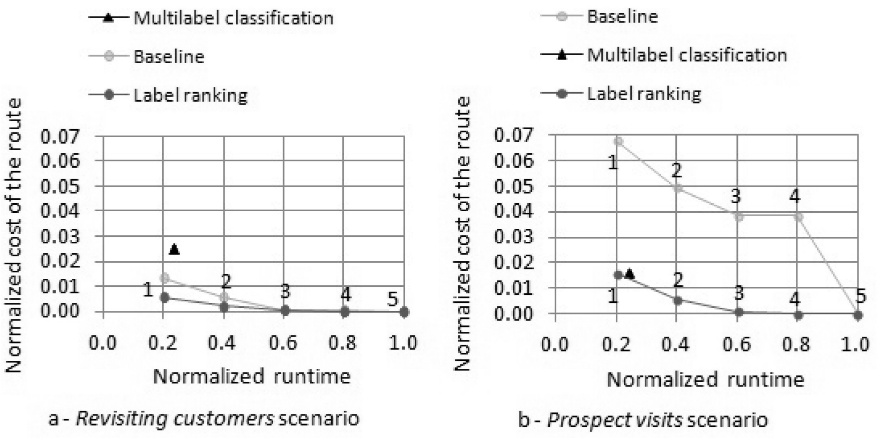


Fig. 2. Normalized average cost of the route versus normalized average runtime for the strategy of running the Top-N MHs for two real-world scenarios

It is hard for the meta-learning approach to achieve significantly higher accuracy than the baseline when predicting the most frequent rankings of MHs. These rankings are usually the most similar to the average ranking. The gain of meta-learning becomes clear in the least frequent rankings. In the *revisiting customers* scenario, the label ranking approach was better than the baseline on six of the seven rankings that were observed only once. For all the different rankings of MHs observed in the *prospect visits* scenario, the label ranking approach presented $\bar{r_S} > 0.6$, while the baseline model achieved this performance in 43% of those rankings.

Figure 2 shows the results for the Top-N strategy. In both scenarios, the Top-1 MH recommended by label ranking model provided a better solution than the Top-1 MH suggested by baseline model. The results for the *revisiting customers* scenario (Figure 2a) show that it is necessary run the Top-2 MHs indicated by the baseline to obtain a solution as good as those provided only by the Top-1 MH of the label ranking. The main advantage of using meta-learning model is the time required to obtain the solution. The time to run the MH, which is in the top position recommended by the meta-learning model, is 50% lower than the time to run the MHs in the first two positions of the baseline ranking. The average solution of the MHs recommended by the multilabel classification is worse than the solution generated by the Top-1 MH of our model. This is due to the fact that the MHs classified for some instances are not the best ones.

For the *prospect visits* scenario, the strategy of running the Top-1 MH suggested by the proposed approach provided a better solution compared to that obtained after processing the four most promising (Top-4) MHs from the baseline ranking — see Figure 2b. The average solution of the MHs ranked by the baseline model is worst for this scenario, in relation to the previous scenario, due to the increase in the number of different rankings observed in the meta-data. The multilabel classification model recommends MHs whose solutions are as good as those provided by Top-1 MH suggested by the label ranking. However, every MH recommended by multilabel classification must be performed to indicate the solution for a given instance, requiring a longer processing time.

The model based on label ranking allows the user to obtain a good solution by running only the Top-1 MH. In practical situations where the user has enough time to run the other recommended MHs, even better solutions can be obtained.

7 Final Remarks

In this study, we addressed the problem of choosing the best MH for a given instance of the TSP. We use a meta-learning approach, which consists of learning a model that relates the properties of the TSP instances with the performance of MHs. We use an adaptation of the MLP for label ranking. Our results show that it is possible to predict the ranking of MHs and that, by following the recommendations in the rankings, it is possible to obtain good quality solutions when compared to simpler selection strategies. In particular, the comparison with a multilabel classification approach to the same problem additionally shows the advantage of addressing the problem as a label ranking task.

We consider two different scenarios: *re-visiting customers*, in which the new instances which we want to select the algorithms for share cities with the instances in the training meta-data; and *prospect customers*, in which the cities in new instances are new. Our results indicate that the latter type of scenario is harder to learn. This is expected because, in the *re-visiting customers* scenario, we cannot really say that the test instances are independent from the training instances because they share parts of their structure. However, since both scenarios may be true in practice, our work shows that it is important to investigate them separately. As future work, we will investigate new meta-features, following different approaches, such as adapting subsampling landmarkers [2].

Acknowledgment. The authors acknowledge CAPES, CNPq, FAPESP, FAPEAM and FCT.

References

1. Applegate, D., Bixby, R., Cook, W.: The Traveling Salesman Problem: A Computational Study. Princeton University Press, New Jersey (2006)
2. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning: Applications to Data Mining. Springer, Berlin (2009)
3. Brazdil, P., Soares, C., Costa, J.: Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. Machine Learning 50, 251–257 (2003)
4. Dekel, O., Manning, C.D., Singer, Y.: Log-Linear Models for Label Ranking. In: Advances in Neural Information Processing Systems. MIT Press (2003)
5. Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation 1(1), 53–66 (1997)

6. Feo, T., Resende, M.: Greedy randomized adaptive search procedures. Journal of Global Optimization 6, 109–133 (1995)
7. Fürnkranz, J., Hüllermeier, E., Mencía, E., Brinker, K.: Multilabel classification via calibrated label ranking. Mach. Learn. 73, 133–153 (2008)
8. Gendreau, M., Potvin, J.Y.: Handbook of Metaheuristics, 2nd edn. Springer Publishing Company, Incorporated (2010)
9. Glover, F., Taillard, E., Taillard, E.: A user's guide to tabu search. Annals of Operations Research 41, 1–28 (1993)
10. Goldberg, D., Lingle Jr., R.: Alleles, loci, and the traveling salesman problem. In: International Conference on Genetic Algorithms and Their Applications, pp. 154–159 (1985)
11. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)
12. Haykin, S.: Neural networks and learning machines, 3rd edn. Pearson Education Inc., New York (2009)
13. Holland, J.: Genetic algorithms and the optimal allocations of trial. SIAM J. Comp. 2, 88–105 (1973)
14. Kanda, J., Carvalho, A., Hruschka, E., Soares, C.: Selection of algorithms to solve traveling salesman problems using meta-learning. International Journal of Hybrid Intelligent Systems 8(3), 117–128 (2011)
15. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science 220, 671–680 (1983)
16. Papadimitriou, C.H.: The euclidean traveling salesman problem is np-complete. Theoretical Computer Science 4(3), 237–244 (1977)
17. Reinelt, G.: TSPLIB - a traveling salesman problem library. ORSA Journal on Computing 3, 376–384 (1991)
18. Rumelhart, D., Hinton, G., Williams, R.: Learning internal representations by error propagation. In: Rumelhart, D., McClelland, J. (eds.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)
19. Smith-Miles, K., van Hemert, J., Lim, X.Y.: Understanding TSP Difficulty by Learning from Evolved Instances. In: Blum, C., Battiti, R. (eds.) LION 4. LNCS, vol. 6073, pp. 266–280. Springer, Heidelberg (2010)
20. Smith-Miles, K., Lopes, L.: Review: Measuring instance dificulty for combinatorial optimization problems. Comput. Oper. Res. 39(5), 875–889 (2012)
21. Spearman, C.: The proof and measurement of association between two things. American Journal of Psychology 15, 72–101 (1904)
22. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Education, Inc., Boston (2006)
23. Vembu, S., Gärtner, T.: Label ranking algorithms: A survey. In: Fürnkranz, J., Hüllermeier, E. (eds.) Preference Learning, pp. 45–64. Springer, Heidelberg (2011)
24. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artificial Intelligence Review 18, 77–95 (2002)
25. Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1, 67–82 (1997)

Modified Particle Swarm Optimization for Pattern Clustering

Swetha K.P[1] and V. Susheela Devi[2]

[1] Dept. Electrical Engineering
Indian Institute of Science. Bangalore, 560012, India
swetha1288@gmail.com
[2] Dept. Computer Science and Automation
Indian Institute of Science. Bangalore, 560012, India
susheela@csa.iisc.ernet.in

Abstract. Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning and data mining. Clustering is grouping of a data set or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait according to some defined distance measure. In this paper we present the genetically improved version of particle swarm optimization algorithm which is a population based heuristic search technique derived from the analysis of the particle swarm intelligence and the concepts of genetic algorithms (GA). The algorithm combines the concepts of PSO such as velocity and position update rules together with the concepts of GA such as selection, crossover and mutation. The performance of the above proposed algorithm is evaluated using some benchmark datasets from Machine Learning Repository. The performance of our method is better than k-means and PSO algorithm.

Keywords: Data Clustering, Particle Swarm Optimization, Fitness Function, Genetic Algorithms.

1 Introduction

Clustering has emerged as one of the most extensively studied research topics due to its numerous important applications in machine learning, image segmentation, data mining and pattern recognition. Recently many clustering algorithms have been proposed. Among them k-means algorithm is the most popular and widely used algorithm because of its easy implementation and efficiency. Although the k-means algorithm was found to produce good clustering quality in many practical problems, the k-means algorithm has some drawbacks [1] such as the selection of initial cluster centers. Another approach to clustering is the agglomerative and divisive hierarchical clustering techniques. Recently, many evolutionary-based clustering algorithms such as Genetic Algorithms (GA) [6], Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and Simulated Annealing (SA) have been introduced. Furthermore, several combinations of these algorithms were used to generate more powerful optimization capabilities. Though the algorithms were superior in their own way, they were slow in finding the optimal solution.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 496–503, 2012.

In this paper we present the genetically improved PSO algorithm, using genetic operators such as selection, crossover and mutation together with the velocity and position updating functions of the basic PSO to search for the optimal results for clusters. A hybrid PSO has been used by Abdel-Kader [5] where, in every iteration, the data is divided into two halves, one being fed to the PSO and the other to the GA. This is the paper closest to our approach but there are also many differences in the two methods. The best combination of PSO+GA has been determined experimentally by us on benchmark datasets and is compared with the basic PSO.

The remaining part of the paper is organised as follows: Section 2 presents the basic principles of PSO and genetic algorithms and the PSO algorithm for the data clustering problem, Section 3 presents the proposed combination of PSO and GA algorithms, Section 4 reports the experimental results and the performance of the proposed algorithm. Finally, conclusions are discussed in Section 5.

2 PSO and GA

2.1 Standard PSO Algorithm

Particle Swarm Optimization (PSO) was originally designed and introduced by Eberhart and Kennedy [4]. PSO originally intends to graphically simulate the graceful and unpredictable choreography of a bird flock. A swarm of computational elements, called particles, is used to explore the solution space for an optimum solution. Each individual within the swarm represents a candidate solution in multidimensional search space and is represented by a vector. The velocity vector is used to determine the next position of the candidate solution. The PSO determines how to update the velocity of a particle [3]. Each particle updates its velocity based on current velocity and the best position it has explored so far and also based on the global best position explored by the swarm.

Each particle i maintains the information, x_i is the current position of the particle. v_i is the current velocity of the particle. y_i is the personal best position of the particle. The particles evolve by updating their velocities and positions according to the following equations:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1(t)(y_i(t) - x_i(t)) + c_2 r_2(t)(\hat{y}(t) - x_i(t)) \tag{1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

Here i =(1, 2,. . . , N) where N is the size of the swarm, ω is the inertia weight, which provides the necessary diversity to the swarm by changing the momentum of the particles to avoid the stagnation of particles at the local optima. c_1 and c_2 are social parameters that are bounded between 0 and 2 and are generally known as the acceleration co-efficients for the particles to move about in the solution space and to pull towards the pbest and gbest positions. r_1 and r_2 are two random numbers, with uniform distribution $U[0, 1]$. $\hat{y}$ is the global best position. The velocity is thus calculated based on previous velocity,the cognitive component which is a function of the distance of the particle from its personal best position and the social component which is a function of the distance of the particle from the best particle found thus far (i.e. the best of the personal bests). The personal best position (pbest) of any particle i, can be viewed as the particle's memory and is the best fitness value achieved so far by the particle i. The personal best position of particle i is calculated as

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } F(y_i(t)) \text{ is better than } F(x_i(t+1)) \\ x_i(t+1) & \text{if } F(x_i(t+1)) \text{ is better than } F(y_i(t)) \end{cases}$$

The global best position is the best position for the entire swarm, and is inferred from all the neighbours in the swarm. The aim of the PSO is to find the particle position that results in the best evaluation of a given fitness (objective) function.

2.2 Standard Genetic Algorithm

Genetic algorithms have been developed by John Holland at the University of Michigan. Genetic algorithms are computing algorithms constructed in analogy with the process of evolution. They seem to be useful for searching very general spaces. Based on the survival and reproduction of the fittest, GA continually exploits new and better solutions. GAs have been applied successfully to problems in any fields such as fuzzy logic control, neural networks, expert systems, and scheduling [2] and have showed their merits over traditional optimization methods. For each problem GA codes the solution as a string where each string is known as a chromosome. At the initial stage the set of chromosomes are taken and are subjected to the genetic search operators such as selection, crossover and mutation one after the other in order to generate a new set of chromosomes (particles) such that the quality would be better when compared to the previous generation, here the quality refers to the fitness measured by a specific function. This process is repeated until the termination criterion is met, and the best chromosome of the last generation is reported as the final solution.

2.3 PSO for Data Clustering Problem

Each particle in the PSO has as many elements as the number of data points. Each element maps to one data point and the value of the element gives the cluster to which the element belongs to. If we have data points X = $(x_1, x_2,\ldots, x_N)$ then the i^{th} particle S_i is represented as S_i = $(y_1, y_2,\ldots, y_n)$ where y_j refers to the cluster to which the j^{th} data point x_j belongs to. If we have k clusters, then y_j is a value between 1 and k. A swarm represents a number of candidate solutions(clusterings). The centroid of particle s_i which represents a clustering of the data is given as follows: A clustering which has a lower value of F is a better clustering of the points.

$$m_j = \frac{\sum_{y \in C_j} y}{\mid C_j \mid} \quad j = 1, \ldots, k \tag{3}$$

The fitness function is evaluated for each particle and is compared with its own best previous fitness value and to the best fitness of all the particles in the swarm, the fitness value calculation is as follows:

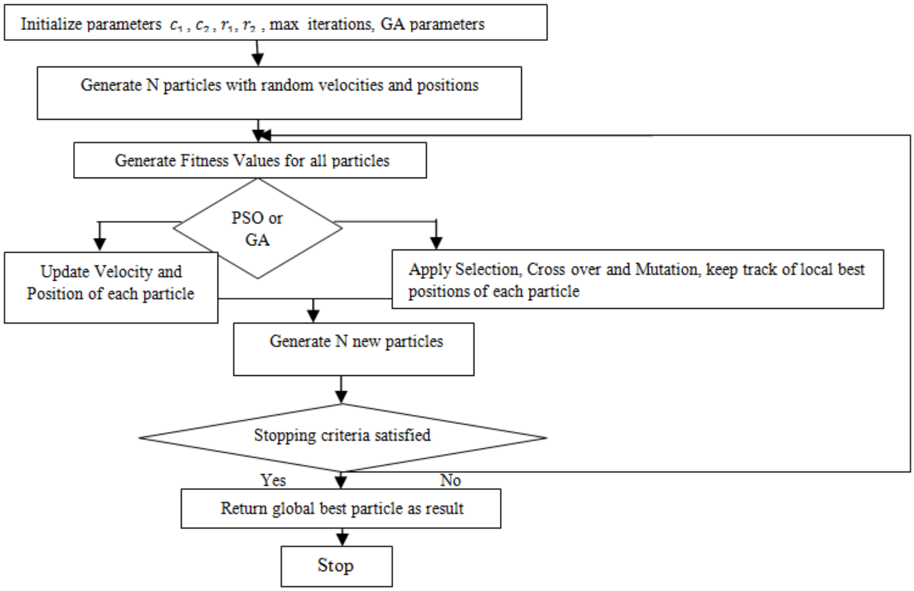

Initialize parameters c_1, c_2, r_1, r_2, max iterations, GA parameters
Generate N particles with random velocities and positions
Generate Fitness Values for all particles
PSO or GA
Update Velocity and Position of each particle
Apply Selection, Cross over and Mutation, keep track of local best positions of each particle
Generate N new particles
Stopping criteria satisfied
Yes
No
Return global best particle as result
Stop

Fig. 1. Flowchart of GA+PSO Algorithm

$$F = \frac{\sum_{j=1}^{k}[\frac{\sum_{\forall y \in C_j} d(y,m_j)}{|C_j|}]}{\sum_{j=1}^{k} d(m_j, m)} \tag{4}$$

and

$$m = \frac{\sum_y y}{n} \tag{5}$$

Where $d(m_j, m)$ gives the distance between the centroid of j^{th} cluster and the centroid of all the points.

3 GA Combined with PSO

GA is known for its randomized search of natural evolution. In the method proposed by us as shown in Fig(1), in every iteration either the entire set of particles are updated using a PSO or the entire set is passed to the GA and the next generation of particles are generated. The population size of the GA-PSO algorithm is set to N. The initial N particles are randomly generated and their fitness function is calculated, these N particles are then fed into the PSO search algorithm. In each iteration, the particle adjusts the vector position in the vector space according to its own experience and those of its neighbours, the fitness function is recalculated, the particles created by PSO are used as the new population. When the particles go into the GA loop, selection, crossover and mutation is carried out to get the new population of particles. When selection is carried out, only the best particle is taken and reproduced in the next generation and there by removing the weak particles, also the local best of each particle is also maintained. When crossover is carried out between two particles, both these particles will keep track of the local best as the better of the two local best positions of the two particles. The fitness function is recalculated again and the process is repeated for maximum number of iterations or until certain convergence criteria are met.

The combined algorithm is as follows:

```
1. Initialize each particle to contain elements, each set randomly to cluster number.
2. For $t = 1$ to $t_{max}$ do
   /* $t_{max}$ is the maximum number of iterations */
   Run step a or step b
   (a) PSO
       i. For each particle S do
          A. Calculate the fitness using (4)
          B. Update local best position
       ii. Update the global best position
       iii. Update the particles using (1) and (2)
   (b) GA
       i. Generate the new generation of particles using selection, crossover and
          mutation keeping track of local best for each new particle generated.
```

4 Experimental Results

The proposed algorithm has been implemented using MATLAB 7.1 and executed on Intel core i3 processor, 2.27 GHz computer. the results are generated and compared on various standard data sets obtained from [7]. In our algorithm called the PSO+GA algorithm, the parameters were set as follows: From the literature done we found that inertia weight ω is set to 0.79 to get a better result [8],The two constant c_1 and c_2 are set to 2 [9]. Deterministic selection procedure is used as the selection function for the GA. Random one-point crossover and mutation are used as the genetic operators where the mutation probability is 0.01 and crossover probability 0.95. k is the number of classes in the data set that is already predefined in the benchmark datasets[7]. The results are compared with the basic Kmeans algorithm and among different combinations where PSO refers to only basic PSO, Hybrid PSO refers to the method proposed by Rehab F. Abdel-Kader [5]. In 1PSO+1GA after every iteration of PSO we do one GA and so on until maximum number of iterations are reached, in 3PSO+1GA, after 3 iterations of PSO we do one GA and so on. For the test problem, the factors such as Mean squared error(MSE), Entropy(E), Purity(P) and the average CPU time are calculated. The four datasets have been used as shown in Table 1. The results have been given for different combinations of PSO and GA in Table 2, 3, 4 and 5. It can be seen that in most cases, the fitness value, MSE and Entropy of our method is lower then that of Kmeans, PSO and Hybrid PSO. Purity is also generally higher in our method. The right combination of

PSO and GA needs to be chosen. This depends on the dataset. We can use a validation dataset to decide which combination of PSO+GA to use.

Here is the summarization of the datasets used as shown in following table:

Table 1. DataSets Used

DataSets	**No of Instances**	**No of Attributes**	**No of Classes**
Iris	150	4	3
Wine	178	13	3
CMC	1473	10	3
Optical	5620	64	10

Table 2. Iris Data with t = 10, N = 10

Methods	Fitness Value	MSE	Entropy	Purity	Avg CPU Time
Kmeans	1.125 ± 0.05	1.023 ± 0.11	1.234 ± 0.16	0.66	0.20s
PSO	0.190 ± 0.21	0.189 ± 0.08	1.169 ± 0.63	0.64	1.24s
Hybrid PSO	1.003 ± 0.66	0.89 ± 0.90	1.091 ± 0.30	0.68	1.82s
1PSO + 1GA	0.176 ± 0.14	0.181 ± 0.35	1.172 ± 0.69	0.80	2.28s
3PSO + 1GA	0.189 ± 0.80	0.251 ± 0.78	1.172 ± 0.80	0.76	2.53s
5PSO + 1 GA	0.266 ± 0.10	0.258 ± 0.25	1.210 ± 0.15	0.74	2.73s

Table 3. Wine Data with t = 10, N = 10

Methods	Fitness Value	MSE	Entropy	Purity	Avg CPU Time
Kmeans	1.102 ± 0.05	1.025 ± 0.11	1.365 ± 0.16	0.65	0.65s
PSO	0.282 ± 0.21	0.263 ± 0.08	1.321 ± 0.63	0.58	1.82s
Hybrid PSO	0.235 ± 0.66	0.172 ± 0.90	1.356 ± 0.30	0.56	2.82s
1PSO + 1GA	0.163 ± 0.14	0.150 ± 0.35	1.299 ± 0.69	0.73	3.01s
3PSO + 1GA	0.170 ± 0.80	0.158 ± 0.78	1.315 ± 0.80	0.63	3.53s
5PSO + 1 GA	0.181 ± 0.10	0.162 ± 0.25	1.320 ± 0.15	0.61	3.73s

Table 4. CMC Data with t = 10, N = 10

Methods	Fitness Value	MSE	Entropy	Purity	Avg CPU Time
Kmeans	1.125 ± 0.05	1.023 ± 0.11	1.234 ± 0.16	0.66	0.20s
PSO	0.190 ± 0.21	0.189 ± 0.08	1.169 ± 0.63	0.64	1.24s
Hybrid PSO	1.003 ± 0.66	0.89 ± 0.90	1.091 ± 0.30	0.68	1.82s
1PSO + 1GA	0.176 ± 0.14	0.281 ± 0.35	1.063 ± 0.69	0.80	2.28s
3PSO + 1GA	0.489 ± 0.80	0.512 ± 0.78	1.172 ± 0.80	0.76	2.53s
5PSO + 1 GA	0.566 ± 0.10	0.580 ± 0.25	1.210 ± 0.15	0.74	2.73s

Table 5. Optical Recognition Data with t = 15, N = 20

Methods	Fitness Value	MSE	Entropy	Purity	Avg CPU Time
Kmeans	3.003 ± 0.05	2.270 ± 0.11	2.641 ± 0.16	0.70	612.77s
PSO	2.001 ± 0.21	2.188 ± 0.08	2.300 ± 0.63	0.68	758.23s
Hybrid PSO	2.121 ± 0.66	2.150 ± 0.90	2.210 ± 0.30	0.70	957.2s
1PSO + 1GA	1.882 ± 0.14	1.771 ± 0.35	1.856 ± 0.69	0.81	950.23s
3PSO + 1GA	2.004 ± 0.80	1.993 ± 0.78	1.953 ± 0.80	0.72	959.58s
5PSO + 1 GA	2.158 ± 0.10	2.005 ± 0.25	2.100 ± 0.15	0.70	1005.4s

5 Conclusions

In this paper, we have proposed a genetically improved PSO where genetic operators are introduced after some of the PSO iterations. The results show that our proposed method gives consistently better results. Though the time taken is more,it is only slightly higher and worth using since better clustering is obtained.

References

1. Ahmad, A., Dey, L.: A k-mean clustering algorithm for mixed numeric and categorical data. Data & Knowledge Engineering 63, 503–527 (2007)
2. Maulik, U., Bandyopadhyay, S., Sanghamitra, B.: Genetic Algorithm-Based Clustering Technique. Pattern Recognition 33, 1455–1465 (2000)

3. Van der Merwe, D.W., Engelbrecht, A.P.: Data Clustering using Particle Swarm Optimization. In: The Congress on Computational Intelligence, Evolutionary Computation, CEC 2003, vol. 1, pp. 215–220 (2003)
4. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)
5. Kader, A., Rehab, F.: Genetically Improved PSO Algorithm for Efficient Data Clustering. In: Proceedings of the 2010 Second International Conference on Machine Learning and Computing, vol. 10, pp. 71–75 (2010)
6. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning, pp. 60–85 (2009)
7. UCI Repository of Machine Learning Databases, http://www.ics.uci.edu/~mlearn/MLRepository.html
8. Eberhart, R.C., Shi, Y.H.: Particle Swarm Optimization: Developments, Applications and Resources. In: Evolutionary Computation, vol. 1, pp. 81–86 (2001)
9. Hu, X.L., Shi, Y.H., Eberhart, R.: Recent Advances in Particle Swarm. In: Evolutionary Computation, CEC, vol. 1, pp. 90–97 (2004)

A Weighted Learning Vector Quantization Approach for Interval Data

Telmo M. Silva Filho and Renata Maria Cardoso R. de Souza

Universidade Federal de Pernambuco, Centro de Informatica,
Av. Prof. Luiz Freire, s/n, 50740-540 Recife (PE), Brazil
{tmsf,rmcrs}@cin.ufpe.br
http://www.cin.ufpe.br/

Abstract. Symbolic Data Analysis deals with complex data types, capable of modeling internal data variability and imprecise data. This paper introduces a Learning Vector Quantization algorithm for symbolic data that uses a weighted interval Euclidean distance to try and achieve a better performance of classification when the dataset is composed of classes of varying structures. This algorithm is compared to a Learning Vector Quantization algorithm that uses traditional interval Euclidean distance. The algorithms are evaluated and compared for their performances with synthetic and real datasets. This paper aims at contributing to the area of Supervised Learning within Symbolic Data Analysis.

Keywords: Symbolic Data Analysis, Interval Data, Supervised Learning, Learning Vector Quantization, Weighted Distance.

1 Introduction

The interest in interval data has grown with the recent advances in database and computational intelligence technologies. This type of data has been mainly studied in *Symbolic Data Analysis* (SDA) [1], which is a domain in the area of knowledge discovery and data management, related to multivariate analysis, pattern recognition and artificial intelligence.

Several supervised classification tools have been extended to handle interval data. Rossi and Conan-Guez [2] have generalized Multi-layer Perceptrons to work with interval data. Appice et al. [3] introduced a lazy-learning approach that extends a k-Nearest Neighbor with weighted distance to interval and modal data. Silva and Brito [4] proposed three approaches to the multivariate analysis of interval data, focusing on linear discriminant analysis.

One algorithm that shows great potential for being extended to handle interval data, due to its simplicity and efficiency, is *Learning Vector Quantization* (LVQ). LVQ is a prototype-based algorithm proposed by Kohonen [5]. The algorithm starts by randomly assigning a subset of prototypes for each pattern class of the dataset. Then, the prototypes are iteratively updated such that the nearest neighbor rule minimizes the average expected misclassification probability.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 504–511, 2012.

When the iterations stop, the updated prototypes should be close to the training patterns in their classes.

Several modifications of the basic LVQ procedure have been proposed. Paredes and Vidal [6] proposed an LVQ algorithm with prototype reduction which had local weights for its prototypes. Silva Filho and Souza [7] introduced an LVQ algorithm which used a prototype-based Euclidean distance to better model classes composed of subregions of different shapes.

The main contribution of this paper is to introduce an LVQ classifier for interval data which uses an interval adaptation of the prototype-based Euclidean distance introduced by Silva Filho e Souza. The prototype-based distance allows for better modeling of classes which have subregions with different shapes in their structures.

The paper is organized as follows: Section 2 further explains interval symbolic data and its characteristics, introducing the synthetic and real datasets used in this paper. Section 3 introduces the LVQ with a prototype-based interval Euclidean distance. Section 4 presents a performance analysis –comparing the algorithm presented at Section 3 and an LVQ which uses non-weighted interval Euclidean distance– considering the datasets presented at Section 2. And lastly Section 5 gives the final remarks.

2 Symbolic Data

Classic data patterns are usually defined as vectors of quantitative or qualitative variables. Due to this fact, classic data analysis does not naturally comprehend variability or uncertainty for the representation of complex data. Symbolic Data Analysis introduces a number of data types that better represent data variability, e.g. intervals, histograms, lists of values, and others [8].

This paper focuses on interval data. This data type comes naturally from the description of ranges of values, e.g. daily temperature variation, daily stock prices, high and low water values in a tide table, etc. Interval data can also help dealing with imprecise data.

Suppose there are K classes labeled $1, \ldots, K$. Let $\Im = \{(\mathbf{x}_i, y_i)\}$ $(i = 1, \ldots, N)$ be a symbolic learning dataset. Each item i is described by a vector of p symbolic variables $\mathbf{x}_i = (X_i^1, \ldots, X_i^p)$ and a discrete quantitative variable Y that takes values in discrete set $G = \{1, \ldots, K\}$. A symbolic variable X_i^j $(j = 1, \ldots, p)$ is an interval-valued variable when, given an item i of $\Im$, $X_i^j = [L_i^j, U_i^j] \subseteq \mathcal{A}_j$ where $\mathcal{A}_j = [L, U]$ is an interval. Both datasets used in this paper are composed of interval-valued variables.

2.1 Synthetic Interval Dataset

In order to generate a synthetic interval dataset, we must first generate a quantitative dataset. Each subregion in this quantitative dataset was drawn according to a bi-variate Gaussian distribution with non-correlated components. Each data point (z_1, z_2) of this synthetic quantitative dataset is a seed of a vector of intervals (a hypercube of p dimensions, where p is the number of variables) defined as: $[([z_1 - \phi_1/2, z_1 + \phi_1/2], [z_2 - \phi_2/2, z_2 + \phi_2/2])]$. These parameters ϕ_1, ϕ_2 are randomly selected from the same predefined generating interval. The generating intervals considered in this paper are: $[1, 30]$ and $[1, 50]$.

The generated interval dataset has 200 rectangles scattered among two classes with 100 rectangles for each class. The figure below shows that each class has 2 differently shaped subregions and the classes show some overlapping. The parameters for the dataset are:

1. Class 1:
   - Subregion 1: $\mu_1 = 165$, $\mu_2 = 185$, $\sigma_1^2 = 196$, $\sigma_2^2 = 1521$ and $n = 50$.
   - Subregion 2: $\mu_1 = 227$, $\mu_2 = 263$, $\sigma_1^2 = 324$, $\sigma_2^2 = 324$ and $n = 50$.
2. Class 2:
   - Subregion 1: $\mu_1 = 195$, $\mu_2 = 187$, $\sigma_1^2 = 196$, $\sigma_2^2 = 1521$ and $n = 50$.
   - Subregion 2: $\mu_1 = 120$, $\mu_2 = 122$, $\sigma_1^2 = 324$, $\sigma_2^2 = 324$ and $n = 50$.

The following figure shows the generated interval datasets.

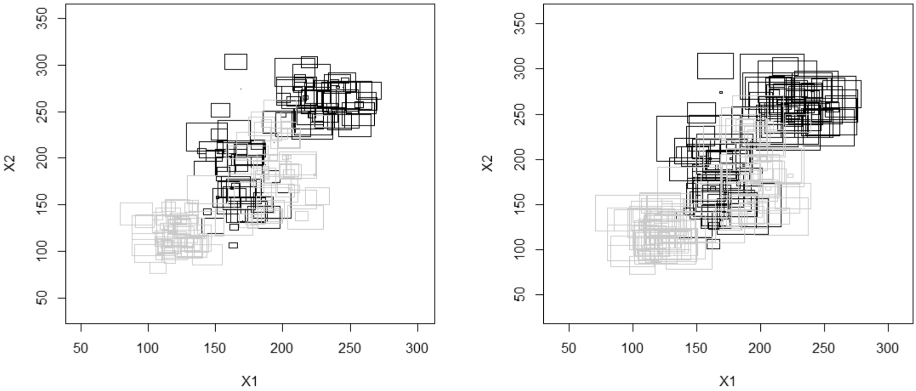


Fig. 1. Dataset considering intervals [1,30] and [1,50]. Classes are represented as groups of rectangles of different colors.

2.2 Dry Climates Dataset

A climate dataset was extracted from a global web site [9] which presents official weather observations, weather forecasts and climatological information for selected cities supplied by National Meteorological & Hydrological Services (*NMHSs*) worldwide. The *NMHSs* make official weather observations in their respective countries.

This dataset has 522 cities with 17 variables of which 16 are interval-valued variables (minimum and maximum temperature in each month and minimum and maximum precipitation in each season) and one is a quantitative variable defining the city's Köppen climate classification [10]. To avoid confusion with the season inversion between northern and southern hemispheres, the first month of summer is considered the first month of the year. The classes are: desert (148 cities), savanna (199 cities) and semi-arid (175 cities). All variables were rescaled by normalization between 0 and 1. This dataset is available online [11].

3 The Learning Vector Quantization with an Interval Prototype-Based Euclidean Distance

The LVQ version extended in this paper is the *Optimized Learning Vector Quantization* (an online LVQ with distinct learning rates for each prototype) [5]. The fact that *Optimized Learning Vector Quantization* (OLVQ) is a prototype based algorithm makes adapting it from quantitative to interval data a simple task. It is a matter of changing the distance used to find the winner prototype from a classic distance to an interval distance, e.g. an interval Euclidean distance.

The formula for interval Euclidean distance is

$$d_{int}(\mathbf{x}_i, \mathbf{w}_m) = d_E^L(\mathbf{x}_i, \mathbf{w}_m) + d_E^U(\mathbf{x}_i, \mathbf{w}_m) \tag{1}$$

where $d_E^L(\mathbf{x}_i, \mathbf{w}_m)$ and $d_E^U(\mathbf{x}_i, \mathbf{w}_m)$ are, respectively, the classic Euclidean distance between the lower and upper values of the interval-valued variables of pattern i and prototype m. The OLVQ applied to interval data, using the interval Euclidean distance is called *Interval Learning Vector Quantization* (ILVQ).

ILVQ and other prototype-based methods which use Euclidean distance to find the nearest prototype tend to work better with spherical clusters and/or classes. This is because, when using Euclidean distance, points that are equally distant to a prototype are positioned in a radius around it.

To allow ILVQ to model classes composed of subregions of varying shapes, we extend the prototype-based Euclidean distance proposed by Silva Filho and Souza to interval data.

3.1 Interval Prototype-Based Euclidean Distance

Suppose there are M prototypes, such that each prototype $m \in (1, \dots, M)$, is described by a vector of p interval-valued variables $\mathbf{w}_m = (W_m^1, \dots, W_m^p)$ and a discrete quantitative variable Y that takes values in discrete set $G = \{1, \dots, K\}$. When using the interval prototype-based Euclidean distance, each prototype m must have a weight vector λ_m, such that each of its interval valued variables W_m^j has a different weight λ_m^j.

The weighted interval Euclidean distance of prototype m at instant t ($dp_{m(t)}$) is calculated as follows:

$$dp_{m(t)}(\mathbf{x}_i, \mathbf{w}_m) = d_{m(t)}^L(\mathbf{x}_i, \mathbf{w}_m) + d_{m(t)}^U(\mathbf{x}_i, \mathbf{w}_m) \tag{2}$$

where $d^L_{m(t)}(\mathbf{x}_i, \mathbf{w}_m)$ and $d^U_{m(t)}(\mathbf{x}_i, \mathbf{w}_m)$ are, respectively, the classic prototype-based Euclidean distance between the lower and upper values of the interval-valued variables of pattern i and prototype m. This classic prototype-based Euclidean distance is calculated as follows:

$$d_{m(t)}(\mathbf{x}_i, \mathbf{w}_m(t)) = \sqrt{\sum_{j=1}^{p} \lambda^j_m (x^j_i - w^j_m(t))^2} \tag{3}$$

The weight vector λ_m models the dispersion of the class subregion represented by prototype m. From the Lagrange Multiplier method, the weight vectors $\lambda_m = (\lambda^1_m, \ldots, \lambda^p_m)$ $(m = 1, \ldots, M)$, which follow two restrictions ($\lambda^j_m > 0$ and $\prod_{j=1}^{p} \lambda^j_m = 1$), are updated according to the following expression:

$$\lambda^j_m = \frac{\{\prod_{h=1}^{p}(\Delta^h_m)\}^{\frac{1}{p}}}{\Delta^j_m} \tag{4}$$

where $h = 1, \ldots, p$ is the set of indexes of the variables and Δ^j_m is the sum of the quadratic differences between the lower and upper values of prototype m and the patterns correctly affected by it and is calculated as follows:

$$\Delta^j_m = \sum_{i \in m} [(\mathbf{x}^{jL}_i - \mathbf{w}^{jL}_m)^2 + (\mathbf{x}^{jU}_i - \mathbf{w}^{jU}_m)^2]\delta_{mi} \tag{5}$$

where $\delta_{mi} = 1$ if i belongs to the same class as m, otherwise $\delta_{mi} = 0$. If $\Delta^j_m = 0$ at any iteration of the algorithm, then no updates are made at this iteration.

3.2 The Weighted Interval Learning Vector Quantization Algorithm

Using the interval prototype-based Euclidean distance, we have a *Weighted Interval Learning Vector Quantization* (WILVQ). Since this is an online algorithm, the sum Δ^j_m defined at equation (5) must be updated every time a prototype m affects a training pattern i of the same class. Then, the weight vector λ_m must be recalculated.

To address the problem of information outdating on the Δ^j_c sum (where c is the index of the winning prototype), the update of Δ^j_c is a weighted sum of the new sum of the quadratic differences between the lower and upper values of the new training pattern i and its winning prototype $c(t)$ and the previous $\Delta^j_c(t-1)$, using the learning rate $\alpha_c(t)$ of the winning prototype $c(t)$ as weight, which gives:

$$\Delta^j_c = [(1 - \alpha_c(t))\Delta^j_c(t-1)] + \{\alpha_c(t)[(\mathbf{x}^{jL}_i - \mathbf{w}^{jL}_c)^2 + (\mathbf{x}^{jU}_i - \mathbf{w}^{jU}_c)^2]\} \tag{6}$$

The index c of the nearest prototype to a pattern i is found as follows:

$$c = arg\,min\{dp_m(\mathbf{x}_i, \mathbf{w}_m) \forall m \in (1, \ldots, M)\} \tag{7}$$

The steps for WILVQ are presented below.

1. **Initialization**
   1.1 At instant $t = 0$ Choose the M prototypes $\{(\mathbf{w}_1(t), y_1), \ldots, (\mathbf{w}_M(t), y_M)\}$.
   1.2 From $m = 1$ to M, do $\alpha_m(t) = 0.3$ and, from $j = 1$ to p, do $\lambda_m^j = 1$.
2. **Prototype update step:** choose a pattern i of the training dataset $\Im$ randomly.
   2.1 Define the winning prototype c using Equation (7)

   2.2 If class(i) = class(c) do
      2.2.1 Update $\Delta_c^j, j \in (1, \ldots, p)$ using Equation (6).

      2.2.2 Update the weight vector λ_c using Equation (4).

      2.2.3 From $j = 1$ to p, do $W_c^j(t+1) = W_c^j(t) + \alpha_c(t)[X_i^j - W_c^j(t)]$, where the subtraction of intervals X_i^j and $W_c^j(t)$ is intuitively calculated as $X_i^j - W_c^j(t) = [L_i^j - L_c^j, U_i^j - U_c^j]$, which means it is the interval formed by the subtraction of the boundaries of the original intervals.

   2.3 If class$(i) \neq$ class(c), then, from $j = 1$ to p, do $W_c^j(t+1) = W_c^j(t) - \alpha_c(t)[X_i^j - W_c^j(t)]$.

   2.4 Update $\alpha_c(t)$ according to the following equation:

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + s(t)\alpha_c(t-1)} \tag{8}$$

   where $s(t) = +1$ if i and c belong to the same class, and $s(t) = -1$ if i and c belong to different classes.
3. **Convergence criterion:** If every $\alpha_m(t) \leq 0.00005$ $(m = 1, \ldots, M)$, STOP. If not, if step 2 has been repeated a number of times equal to the length of the prototype set multiplied by the length of the training set then go to step 4, otherwise go to step 2.
4. **Validation**
   4.1 Compute validation error.
   4.2 If the validation error has grown for three consecutive times, STOP, if not go to step 2.

A validation step is used because the algorithm may pass too many times through step 2 without meeting the convergence criterion for the learning rate. Evaluating the validation error allows the algorithm to take as many turns as it needs on step 2 to converge, before the validation error grows three consecutive times.

4 Experiments

To compare the performances of the WILVQ and the ILVQ, experiments were conducted with the synthetic and real datasets presented in Section 2.

4.1 Synthetic Dataset

For this dataset, the methods are evaluated based on the accuracy prediction measured by the error rate of classification. 100 replications of the dataset with identical statistical properties are obtained and, for each one, training (50% of the original dataset), validation and test (both with 25% of the original dataset) sets are randomly generated. The estimated error rate of classification corresponds to the average of the error rates obtained among the 100 replicates of the test set.

In these experiments, 4 prototypes were used for each of the two classes. The table below presents the results of the average and standard deviation of the error rate of classification (in %) for both generating intervals ($[1, 30]$ and $[1, 50]$). To confirm these results, a two-sampled, two-tailed hypothesis test was made for each generating interval, with a significance level $\alpha = 0.05$. For each test the hypothesis were: $H_0 : \mu_{WILVQ} = \mu_{ILVQ}$ and $H_1 : \mu_{WILVQ} \neq \mu_{ILVQ}$. The number of degrees of freedom used was: $min\{n_{WILVQ} - 1, n_{ILVQ} - 1\}$. Since for every sample $n = 100$, the number of degrees of freedom (df) is always 99.

With $\alpha = 0.05$ and $df = 99$, the critical values for the tests are $t_0 > t_{0.025;99} \approx 1.98$ and $t_0 < t_{0.025;99} \approx -1.98$.

Table 1. Error Rate of the Classifiers for the Synthetic Dataset and Statistics for the Hypothesis Tests

Interval	ILVQ	WILVQ	Statistics
[1,30]	18.12 (7.55)	15.2 (6.20)	-2.99
[1,50]	16.9 (6.16)	14.88 (5.96)	-2.35

Since this synthetic interval dataset was made to show some overlapping and subregions of different shapes within classes, it was expected that WILVQ performed better than ILVQ. This is confirmed by all the hypothesis tests for the error rate (the statistics give evidence that the null hypothesis should be rejected and that $\mu_{WILVQ} < \mu_{ILVQ}$)

4.2 Dry Climates Dataset

WILVQ and ILVQ were evaluated with the dry climates dataset presented at Section 2 for their performances of error rate of classification. The methods were evaluated using a Monte Carlo simulation with 10 repetitions, and within each of these repetitions, a 10-fold cross validation was made, yielding a total of 100 error rate results. 5 prototypes were used for each one of the three classes.

The results of the average and standard deviation of the error rate of classification were: 17.51% with a standard deviation of 5.22% for the ILVQ and 15.67% with a standard deviation of 4.88% for the WILVQ.

In the same way that it was done with the synthetic dataset, a two-sampled, two-tailed hypothesis test was made, with a significance level $\alpha = 0.05$. The hypothesis were: $H_0 : \mu_{WILVQ} = \mu_{ILVQ}$ and $H_1 : \mu_{WILVQ} \neq \mu_{ILVQ}$. The resulting statistics was $t_0 = -2.58 < -1.98$. The result of the hypothesis test shows that WILVQ performs better than ILVQ.

5 Conclusion

An interval weighted LVQ approach for interval data has been introduced in this paper: the WILVQ. It uses an interval adaptation of the prototype-based Euclidean distance proposed by Silva Filho and Souza to try and achieve a better performance than the ILVQ, which uses the traditional non-weighted interval Euclidean distance. Since ILVQ uses an interval non-weighted Euclidean distance to find the nearest prototype, it tends to model classes –and their subregions represented by their prototypes– of spherical shape. Using a prototype-based Euclidean distance, WILVQ is able to describe classes with complex intrastructure defined by subregions.

Experiments with synthetic and real datasets have confirmed that WILVQ tends to outperform ILVQ when classes overlap and have subregions that are not well described by non-weighted interval Euclidean distance.

References

1. Bock, H.H., Diday, E.: Analysis of Symbolic Data: Exploratory Methods for Extracting. Statistical Information from Complex Data. Springer (2000)
2. Rossi, F., Conan, G.B.: Multi-layer Perceptron on Interval Data. In: Classification, Clustering and Data Analysis, pp. 427–434 (2002)
3. Appice, A., Amato, D.C., Esposito, F., Malerba, D.: Classification of Symbolic Objects: A lazy Learning Approach. Intelligent Data Analysis 10(4), 301–324 (2006)
4. Silva, A., Brito, P.: Linear Discriminant Analysis for Interval Data. Computational Statistics 21, 289–308 (2006)
5. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer (2001)
6. Paredes, R., Vidal, E.: Learning Prototypes and Distances. A Prototype Reduction Technique Based on Nearest Neighbor Error Minimization. In: International Conference on Pattern Recognition, vol. 3, pp. 442–445 (2004)
7. Silva Filho, T.M., Souza, R.M.C.R.: Pattern Classifiers with Adaptive Distances. In: International Joint Conference on Neural Networks (IJCNN), pp. 1508–1514 (2011)
8. Diday, E., Noirhomme, F.M.: Symbolic Data Analysis and the SODAS Software. Wiley Interscience (2008)
9. World Meteorological Organization (2012), http://worldweather.wmo.int
10. Encyclopaedia Britannica: Koppen Climate Classification (2012), http://goo.gl/U6lq6
11. Silva Filho, T.M.: Dry Climates Dataset, http://goo.gl/aWMBC

A Genetic Algorithm Solution for the Operation of Green LTE Networks with Energy and Environment Considerations

Hakim Ghazzai[1], Elias Yaacoub[2], Mohamed Slim Alouini[1], and Adnan Abu-Dayya[2]

[1] King Abdullah University of Science and Technology (KAUST),
Thuwal, Mekkah Province, Saudi Arabia
{hakim.ghazzai,slim.alouini}@kaust.edu.sa
[2] Qatar Mobility Innovations Center (QMIC),
Qatar Science & Technology Park, Doha, Qatar
{eliasy,adnan}@qmic.com

Abstract. The Base Station (BS) sleeping strategy has become a well-known technique to achieve energy savings in cellular networks by switching off redundant BSs mainly for lightly loaded networks. Besides, the exploitation of renewable energies, as additional power sources in smart grids, becomes a real challenge to network operators to reduce power costs. In this paper, we propose a method based on genetic algorithms that decreases the energy consumption of a Long-Term Evolution (LTE) cellular network by not only shutting down underutilized BSs but also by optimizing the amounts of energy procured from the smart grid without affecting the desired Quality of Service.

Keywords: Green Network, Genetic Algorithm, Sleeping Strategy, Smart Grid.

1 Introduction

Mobile networks represent already around 10% of the total carbon emitted by Information and Communication Technology (ICT) and this is expected to increase every year [1]. Several works focus on strategies to achieve energy savings in the recent 4G LTE by switching off base stations (BSs), mainly during peak-off hours, as they consume more than 50% of the energy due to circuit processing, air conditioning and other factors [2]. Many heuristic algorithms have been proposed to reduce the number of active BSs depending on different criteria based on a certain quality of service (QoS) metric, e.g. [3].

A complementary work is to study the impact of introducing the smart grid which contains different energy sources (e.g., electricity generated from fossil fuels or from renewable energy sources) to power cellular networks [4]. A recent research focuses on the dynamic operation of cellular BSs that depends on the the traffic, real-time pricing provided by the smart grid and the pollutant level associated with the generation of the electricity [5]. However, this work does not consider a particular technology (e.g. LTE) and does not take intercell interference into account.

In this paper, we investigate the performance of a BS sleeping strategy based on a proposed optimization problem by implementing it within a green Genetic Algorithm

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 512–519, 2012.

(GA). In fact, GA is a strong optimization tool used in several applications for LTE networks such as resource allocation [6], and for smart grids as in [7]. In our case, we implement it with the sleeping strategy in order to minimize the energy consumption of the LTE cellular network, reduce the CO_2 emissions in the green LTE cellular network, maximize the profit of the network operator, and maintain a target QoS level.

This is performed given the nature and the cost of the provided energies in the smart grid in addition to the unitary prices of the mobile network operator services. In addition, we take into account both the Uplink (UL) and Downlink (DL) directions, LTE resource allocation, and intercell interference.

The rest of this paper is organized as follows. Section 2 presents the system model. Section 3 describes the problem formulation. The strategy and the proposed green genetic algorithm are detailed in Section 4. Simulation results are presented and analyzed in Section 5. Finally, the conclusions are drawn in Section 6.

2 System Model

We consider a uniform geographical area where an LTE network is deployed and where users are uniformly distributed. The area is divided into cells of equal size where a BS is placed in the center of each cell. In LTE, the available spectrum is divided into Resource Blocks (RB) that contain a fixed number of consecutive subcarriers. RBs are assigned to users according to the resource allocation procedure described in [8] and summarized as follows: each user communicates with a selected BS. Due to the low transmit power of a Mobile Station (MS) compared to the BS transmit power, we associate each user to the BS that offers the best available UL channel gain and from this assigned BS, we allocate the RB that provides the best available DL channel gain to that user.

2.1 Energy Consumption Model for Base Stations

We consider that each BS is equipped with a single omni-directional antenna. The consumed power P_j^{BS} of the j^{th} active BS can be computed as follows [9]:

$$P_j^{BS} = aP_j^{tx} + b, \tag{1}$$

where P_j^{tx} denotes the radiated power of the j^{th} BS. The coefficient a corresponds to the power consumption that scales with the radiated power due to amplifier and feeder losses. The term b models an offset of site power which is consumed independently of the average transmit power and is due to signal processing, battery backup, and cooling.

2.2 Operator Services

In our framework, the network operator offers M different services characterized by their data rate thresholds $R_{m,th}^{(\mathrm{UL})}$ and $R_{m,th}^{(\mathrm{DL})}$ for UL and DL, respectively, and their unitary prices $p^{(m)}$ with $m = 1 \cdots M$. We suppose that each user in the network benefits from one of the M offered services.

2.3 Retailers and Pollutant Levels

In our study, we assume that the cellular network is powered by a smart grid where N retailers exist to provide energy with different prices and pollutant levels depending on the nature of the energy source. The amount of energy $q_j^{(n)}$ procured by the j^{th} BS from each retailer n ($n = 1 \cdots N$) is a function of its cost (i.e. the unitary price of the provided energy) $\pi^{(n)}$ and a penalty term corresponding to pollutant emissions and modeled as follows [10]:

$$F(q_j^{(n)}) = \alpha_n (q_j^{(n)})^2 + \beta_n q_j^{(n)}, \tag{2}$$

where α_n and β_n are the emission coefficient cost of retailer n. In addition, we suppose that each retailer has a maximum available amount of energy. For instance, the network can not procure from the renewable energy retailer more than a certain amount $Q_{\max}^{(n)}$.

Based on this system model and these parameters, we formulate an optimization problem where the mobile network operator is able to optimally procure energy for its BSs in order to maximize an objective function that depends on its attitude towards the environment. More details about the channel model and the data rate expressions for LTE can be found in [8].

3 Problem Formulation

In this section, we formulate the optimization problem that will be solved in section 4 using the GA.

We consider that N_{BS} BSs are deployed and N_U users are randomly distributed in the area of interest. We denote by N_{out} the number of users in outage ($N_{\text{out}} \ll N_U$). A user i using the m^{th} service communicates successfully with a BS, if its UL and DL data rates, denoted $R_i^{(\text{UL})}$ and $R_i^{(\text{DL})}$ respectively, are higher than the service data rate thresholds: $R_{m,th}^{(\text{UL})}$ and $R_{m,th}^{(\text{DL})}$ respectively. We associate a binary parameter γ_i, $i = 1 \cdots N_U$ to each user. If the user i is served successfully then $\gamma_i = 1$ else $\gamma_i = 0$. If we denote $\boldsymbol{\gamma} = [\gamma_1 \cdots \gamma_{N_U}]$, then the number of ones and the number of zeros in $\boldsymbol{\gamma}$ correspond to the number of served users and the number of users in outage, respectively. Therefore, the network operator revenue $\mathcal{R}(\boldsymbol{\gamma}) = \sum_{i=1}^{N_U} \gamma_i p_i^{(m)}$, where $p_i^{(m)}$ is the cost of the service m used by the i^{th} user, depends only on the spending of the served users.

On the other hand, in order to include the BS sleeping strategy in the problem formulation, we introduce a binary variable ϵ_j with $j = 1 \cdots N_{\text{BS}}$ to denote the BS state: $\epsilon_j = 1$ means that BS j is switched on while $\epsilon_j = 0$ indicates that BS j is switched off. Let $\boldsymbol{\epsilon} = [\epsilon_1 \cdots \epsilon_{N_{\text{BS}}}]$. The number of ones and the number of zeros in this vector indicate the number of active and inactive BSs, respectively. Assume that each BS is able to procure energy from different retailers at the same time. Then, the total cost of the energy consumption of the network is expressed as: $\sum_{j=1}^{N_{\text{BS}}} \sum_{n=1}^{N} \epsilon_j \pi^{(n)} q_j^{(n)}$. In addition, the total CO$_2$ emission of the network is given by: $\sum_{j=1}^{N_{\text{BS}}} \sum_{n=1}^{N} \epsilon_j \left(\alpha_n (q_j^{(n)})^2 + \beta_n q_j^{(n)} \right)$, where $\pi^{(n)}$ is the cost of one unit of energy provided by the nth retailer and $q_j^{(n)}$ is the amount of energy procured by BS j from the n^{th} retailer, with $j = 1 \cdots N_{\mathrm{BS}}$ and $n = 1 \cdots N$. In our work, the mobile network operator has to solve the following optimization problem in order to maximize the utility function U:

$$\underset{\boldsymbol{\gamma},\boldsymbol{\epsilon},\boldsymbol{q}}{\text{Maximize}}\ U = (1-\omega)\,\mathcal{P}(\boldsymbol{\gamma},\boldsymbol{\epsilon},\boldsymbol{q}) - \omega\,\mathcal{I}(\boldsymbol{\epsilon},\boldsymbol{q}), \tag{3}$$

$$\text{Subject to: } \sum_{j=1}^{N_{\mathrm{BS}}} \epsilon_j q_j^{(n)} \leq Q_{\max}^{(n)}\ \forall n = 1 \cdots N, \tag{4}$$

$$\sum_{n=1}^{N} q_j^{(n)} = P_j^{BS}\ \forall j = 1 \cdots N_{\mathrm{BS}}, \tag{5}$$

$$\frac{N_{\mathrm{out}}}{N_U} \leq P_{\mathrm{out}}, \tag{6}$$

$$q_j^{(n)} \geq 0\ \forall j = 1 \cdots N_{\mathrm{BS}},\ \forall n = 1 \cdots N, \tag{7}$$

where ω is a weighting parameter. In (3), $\mathcal{P}(\boldsymbol{\gamma},\boldsymbol{\epsilon},\boldsymbol{q})$ is a function that corresponds to the mobile operator's profit. It is given by:

$$\mathcal{P}(\boldsymbol{\gamma},\boldsymbol{\epsilon},\boldsymbol{q}) = \sum_{i=1}^{N_U} \gamma_i p_i^{(m)} - \sum_{j=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N} \epsilon_j \pi^{(n)} q_j^{(n)}. \tag{8}$$

The function $\mathcal{I}(\boldsymbol{\epsilon},\boldsymbol{q})$ in (3) reflects the friendliness to the environment of the mobile network operator and corresponds to the CO_2 emissions caused by the energy consumption of the mobile operator's network. It is given by:

$$\mathcal{I}(\boldsymbol{\epsilon},\boldsymbol{q}) = \sum_{j=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N} \epsilon_j \left(\alpha_n (q_j^{(n)})^2 + \beta_n q_j^{(n)} \right) \cdot \tag{9}$$

Thus, the objective is to solve a multi-objective optimization (or Pareto optimization) problem by constructing a single aggregate objective function [11] which corresponds to a weighted linear sum of the objective functions (8) and (9). These functions are weighted by a parameter ω called the Pareto weight ($0 < \omega < 1$). The elements of the vector $\boldsymbol{q} = [q_1^{(1)} \cdots q_1^{(N)} q_2^{(1)} \cdots\cdots q_{N_{\mathrm{BS}}}^{(N)}]^T$, and the binary vectors $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$ are the decision variables of the problem.

When $\omega \to 0$, we are dealing with a selfish network operator that aims to maximize its own profit $\mathcal{P}$ regardless of its impact on the environment. When $\omega \to 1$, we deal with an environmentally friendly network operator that aims to reduce CO_2 emissions regardless of its own profit. Other values of ω constitute a tradeoff between these two extremes.

The constraint (4) indicates that the power consumed by all BSs in the cellular network from power retailer n cannot exceed the total power provided by that retailer. While (5) indicates that the amount of power drawn by a BS from all retailers should be equal to the power needed for its operation, (6) forces the number of users in outage to be less than the tolerated outage threshold P_{out} and (7) is a trivial constraint expressing the fact that the energy drawn is a positive amount. It should be noted that, when a certain retailer can provide to the network enough electricity to power all its BSs, we can set $Q_{\max}^{(n)} = +\infty$ to relax the constraint (4) for that retailer, although in practice the amount of energy produced is naturally finite.

4 Green Genetic Algorithm

The formulated optimization problem in section 3 is considered as a combinatorial problem due to the existence of binary variables (γ_i and ϵ_j) as decision variables which makes the optimal and exact solution of this nonlinear optimization problem difficult or even impossible to find [11]. Therefore, we employ the heuristic GA.

The idea of the GA is to find the optimal binary string $\boldsymbol{\epsilon}$ that maximizes the utility function expressed in (3). Initially, the GA generates L binary strings of length N_{BS} forming a set called initial population $\mathcal{S}_0$. For each element of $\mathcal{S}_0$, $\boldsymbol{\epsilon}^{(l)},\ l = 1 \cdots L$ which corresponds to a random combination of $\epsilon_j,\ j = 1 \cdots N_{BS}$, the algorithm computes the data rates of all users and compares them to the data rate thresholds after applying the resource allocation algorithm described in [8]. By this way, it identifies the users in outage and consequently the value of the vector $\boldsymbol{\gamma}^{(l)}$. Then, for fixed $\boldsymbol{\epsilon}^{(l)}$ and $\boldsymbol{\gamma}^{(l)}$, the problem in (3) depends only on one decision variable, the vector $\boldsymbol{q}^{(l)}$, and hence becomes a quadratic concave optimization problem that has a unique solution. Finding that solution determines the power to be procured from the available retailers, and hence the utility function can be computed.

After computing L utilities U_l corresponding to each $\boldsymbol{\epsilon}^{(l)}$, we select the L_b highest utility strings ($L_b < L$) on which we apply crossovers and mutations to generate a new population S_1. This procedure is repeated until reaching convergence or until a maximum number of populations is used. Details of the proposed method using the GA are given in the following:

```
– Step 0: Compute the utility function $U_0$ for $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}^{(0)} = [1 \cdots 1])$ and set $U_{\mathrm{m}} = U_0$
  and $\boldsymbol{\epsilon}_{\max} = \boldsymbol{\epsilon}^{(0)}$.
– Step 1: Generate an initial population $\mathcal{S}$ composed of $L$ random $\boldsymbol{\epsilon}^{(l)},\ l = 1 \cdots L$.
– Step 2: while (Not converged) and (maximum number of populations not reached)
    • for $l = 1 \cdots L$
        * Allocate resources (select serving BS and UL and DL RBs) to all users and
          compute $\boldsymbol{\gamma}^{(l)}$ and $N_{\mathrm{out}}^{(l)}$ corresponding to the string $\boldsymbol{\epsilon}^{(l)} \in \mathcal{S}$.
        * if $\frac{N_{\mathrm{out}}^{(l)}}{N_U} \leq P_{\mathrm{out}}$, find $\tilde{\boldsymbol{q}}^{(l)}$ by solving the quadratic optimization problem in
          (3) that results from fixing $\boldsymbol{\epsilon}^{(l)}$ and $\boldsymbol{\gamma}^{(l)}$, then compute the corresponding
          utility $U_l$.
        * else $\boldsymbol{\epsilon}^{(l)}$ is not a suitable solution (we set $U_l = -\infty$). end if
          end for
    • Set $U_{\max} = \max_{l} U_l$ and $\boldsymbol{\epsilon}_{\max} = \boldsymbol{\epsilon}_{l_m}$ where $l_m$ indicates the index of the string
      $\in \mathcal{S}$ that results in the highest utility.
    • Maintain the best $L_b$ strings $\in \mathcal{S}$ to the next population and from them, generate
      $L - L_b$ new strings by applying crossovers and mutations to form a new
      population $\mathcal{S}$.
      end while
```

Convergence is reached when $U_{\max}$ remains constant for several successive iterations. At the end, the optimal BS combination is $\boldsymbol{\epsilon}_{\max}$.

5 Results and Discussion

In this section, we analyze the performance of the sleeping strategy used with the GA presented in Section 4 versus two parameters: the network operator attitude ω and the number of subscribers N_U.

We consider a $5 \times 5\,(\mathrm{Km}^2)$ LTE coverage area with uniform user distribution where N_{BS} BSs are placed uniformly according to the cell radius, selected to be 0.5 km. The LTE and channel parameters are selected as in [8]. All BSs and all MSs have the same power model and the same maximal transmit power, respectively. The outage probability P_{out} is fixed to 2%. These parameters are detailed in Table 1.

In addition, we suppose that the network operator offers $M = 4$ different services. Each one is characterized by its cost (unitary price) $p^{(m)}$ expressed in Monetary Units (MU), DL and UL data rate thresholds, and the occurrence probability of the service as shown in Table 2. The occurrence probability of a given service corresponds to the percentage of subscribers in the network using that service.

Concerning the energy providers, we assume that $N = 3$ retailers produce energy from different sources. Each type of energy source n is characterized by its unitary price $\pi^{(n)}$ MU, total available energy $Q^{(n)}_{\max}$ and two pollutant coefficients α_n and β_n as shows Table 2. We suppose that the second energy provider has a limited amount of power $Q^{(2)}_{\max}$ to be provided to the mobile network: for instance, it can correspond to a renewable energy provider producing electricity from wind or solar energy. The third retailer, in this scenario, produces energy with a very cheap price but it causes a harmful impact on the environment. The $Q^{(2)}_{\max}$ is kept as a variable to investigate its effect on the system performance.

The GA is applied under the following settings: from a population of size $L = 32$, we run the algorithm at most 35 times. From each population, we select $L_b = 0.5L$ strings to the next population while the remaining $0.5L$ strings are obtained by randomly crossing over the L_b strings. The crossover point is, also, chosen randomly between the $0.2L$ and $0.8L$ positions. The mutation probability is set to 0.01.

The performance of the green GA is studied versus the Pareto parameter ω and the number of subscribers N_U. We run the GA by starting from different populations, and we consider the averaged results over several channel realizations and user locations. Numerical results are obtained for three values of ω: the lowest one corresponds to a selfish network operator while the highest ω refers to an environmentally friendly operator as displays Table 3. In addition, we compare the traditional case where 25 BSs are deployed in the area of interest to the proposed algorithm (sleeping strategy with GA). When ω increases, we notice that we are able to reduce the CO_2 emissions by more than 95% thanks to the exploitation of the renewable energy after switching

Table 1. Power and Bandwidth Parameters

Parameter	Value	Parameter	Value	Parameter	Value
$(B^{(\mathrm{DL})}, B^{(\mathrm{UL})})$ (MHz)	$(10, 10)$	$(N^{(\mathrm{DL})}_{\mathrm{RB}}, N^{(\mathrm{UL})}_{\mathrm{RB}})$	$(50, 50)$	MS Tx power (W)	0.125
BS Tx power (W)	10	a	7.84	b (W)	71.5

Table 2. Service and Energy Provider Parameters

Services	Ser. 1	Ser. 2	Ser. 3	Ser. 4
$p^{(m)}$(MU)	10	5	3	1
$R_{m,th}^{(\mathrm{DL})}$ (kbps)	1000	384	256	64
$R_{m,th}^{(\mathrm{UL})}$ (kbps)	384	384	56	64
Occurrence (%)	10	10	30	50

Retailers	Ret. 1	Ret. 2	Ret. 3
$\pi^{(n)}$(MU)	0.05	0.5	0.01
$Q_{max}^{(n)}$ (W)	$+\infty$	300	$+\infty$
α_n	0.02	0	0.5
β_n	0.2	0	0.5

Table 3. Numerical Results

		$N_U = 50$			$N_U = 200$		
	ω	0	0.1	0.5	0	0.1	0.5
Traditional Scenario	Profit (*MU*)	92.62	53.29	-83.34	548.09	471.81	340.51
	CO_2 emissions (*tonne/year*)	22266	3076	2216	89562	3850	2875
	Consumed power (*W*)	1866	1866	1866	2100	2100	2100
	Power from Retailer 1 (%)	0	96	81	0	96	82.72
	Power from Retailer 2 (%)	0	0	16	0	0	14.28
	Power from Retailer 3 (%)	100	4	3	100	4	3
Genetic Algorithm + Sleeping Strategy	Profit (*MU*)	139.72	133.49	63.71	564.12	537.44	405.41
	CO_2 emissions (*tonne/year*)	2864	265	47	27340	608	112
	Consumed power (*W*)	186.45	166.81	166.36	622.92	381.38	368.56
	Power from Retailer 1 (%)	0	96	8.7	0	95	18.6
	Power from Retailer 2 (%)	0	0	91.3	0	1	81
	Power from Retailer 3 (%)	100	4	0	100	4	0.4
	Active BSs	4	3	3	8	5	5

off redundant BSs. Indeed, the percentage of the procured renewable energy from the smart grid reaches more than 81% of the total consumed power which is exactly equal to the available renewable energy $Q_{max}^{(2)} = 300$ W. This is due to the reduced number of active BSs comparing to the traditional case: it goes from 25 to 5 for $N_U = 200$. In addition, for all values of ω, the proposed method can significantly decrease the total energy consumption of the network and thus maximizes the mobile operator's profit by optimally allocating power from the smart grid to the active BSs. However, an environmentally friendly attitude leads to a loss in terms of profit because of the high unitary price of renewable energy provided by the second retailer. That's why, the network operator may use a tradeoff attitude where it can find an acceptable profit with reduced CO_2 emissions; this point can correspond to $\omega = 0.1$, for this scenario. Finally, numerical results show that, after applying the proposed method, a lower ω corresponds to a higher number of active BSs. This is explained by the fact that the network operator needs to activate some additional BSs to serve more subscribers and hence maximize its profit. In the other cases, the network operator prefers to shut down some BSs instead of serving users if the constraint (6) is still satisfied, since the utility would be increasingly affected by the penalty term $\mathcal{I}(\boldsymbol{\epsilon}, \boldsymbol{q})$ in (3).

6 Conclusions

In this paper, we formulated an optimization problem to allocate the energy procured from the smart grid where renewable energy sources are available in order to reduce CO_2 emissions and/or maximize the profit of an LTE mobile operator depending on its attitude towards the environment. The BS sleeping strategy was implemented through a proposed green genetic algorithm that achieved significant energy savings for the investigated LTE network without affecting the required QoS.

Acknowledgments. This work was made possible by NPRP grant # 4-353-2-130 from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

1. Fettweis, G.P., Zimmermann, E.: ICT energy consumption - Trends and challenges. In: 11th International Symposium on Wireless Personal Multimedia Communications (2008)
2. Louhi, J.: Energy efficiency of modern cellular base stations. In: 29th International Telecommunications Energy Conference (INTELEC), pp. 475–476 (2007)
3. Xiang, L., Pantisano, F., Verdone, R., Ge, X., Chen, M.: Adaptive traffic load-balancing for green cellular networks. In: IEEE PIMRC (2011)
4. Samadi, P., Mohsenian-Rad, A., Schober, R., Wong, V., Jatskevich, J.: Optimal real-time pricing algorithm based on utility maximization for smart grid. In: IEEE SmartGridComm, pp. 415–420 (2010)
5. Bu, S., Yu, F.R., Cai, Y., Liu, P.: When the smart grid meets energy-efficient communications: Green wireless cellular networks powered by the smart grid. IEEE Trans. on Wireless Communications (published online, 2012), doi:10.1109/TWC.2012.052512.111766
6. Yang, X., Wang, Y., Zhang, D., Cuthbert, L.: Resource allocation in LTE OFDMA systems using genetic algorithm and semi-smart antennas. In: IEEE WCNC (2010)
7. Ramaswamy, P., Deconinck, G.: Relevance of voltage control, grid reconfiguration and adaptive protection in smart grids and genetic algorithm as an optimization tool in achieving their control objectives. In: IEEE International Conference on Networking, Sensing and Control, ICNSC (2011)
8. Yaacoub, E.: Performance study of the implementation of green communications in LTE networks. In: International Conference on Telecommunications, ICT (2012)
9. Richter, F., Fehske, A., Fettweis, G.: Energy efficiency aspects of base station deployment strategies for cellular networks. In: IEEE VTC-Fall (2009)
10. Senthil, K., Manikandan, K.: Improved tabu search algorithm to economic emission dispatch with transmission line constraint. Int'l J. of Computer Science and Comm. 1, 145–149 (2010)
11. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)

Robust Hypersurface Fitting Based on Random Sampling Approximations

Jun Fujiki[1], Shotaro Akaho[2], Hideitsu Hino[3], and Noboru Murata[3]

[1] Fukuoka University
[2] National Institute of Advanced Industrial Science and Technology
[3] Waseda University
fujiki@fukuoka-u.ac.jp, s.akaho@aist.go.jp
hideitsu.hino@toki.waseda.jp, noboru.murata@eb.waseda.ac.jp

Abstract. This paper considers $N-1$-dimensional hypersurface fitting based on L_2 distance in N-dimensional input space. The problem is usually reduced to hyperplane fitting in higher dimension. However, because feature mapping is generally a nonlinear mapping, it does not preserve the order of lengthes, and this derives an unacceptable fitting result. To avoid it, JNLPCA is introduced. JNLPCA defines the L_2 distance in the feature space as a weighted L_2 distance to reflect the metric in the input space. In the fitting, random sampling approximation of least k-th power deviation, and least α-percentile of squares are introduced to make estimation robust. The proposed hypersurface fitting method is evaluated by quadratic curve fitting and quadratic curve segments extraction from artificial data and a real image.

Keywords: L$_\alpha$PS, L$_k$PD, JNLPCA, RANSAC, fitting.

1 Introduction

Understanding of the structure of data by dimensionality reduction is a fundamental and important task in data processing. One of the geometrical meanings of dimensionality reduction is fitting hyperplane and/or hypersurface to observed data. To extract linear structure of the data, a hyperplane is fit to the data and *principal component analysis* (*PCA*) is commonly used. To extract nonlinear structure, the PCA is extended to many kinds of *nonlinear PCA* (*NLPCA*) and this means many kinds of hypersurface fitting methods are proposed. The basic idea of NLPCA is that the data which have nonlinear structure is mapped to a high-dimensional space, called feature space, so as to have linear structure in the feature space. Then, original PCA is applied to extract linear structure of the data in the feature space. In the framework of NLPCA, type of extractable nonlinear structure strongly depends on this nonlinear mapping, which is called feature mapping. Hence, the feature mapping determines the class of fitting hypersurface and selecting an appropriate feature mapping is very important. In many applications in computer vision, the type of structure, that is, the class of hypersurface is known such as quadratic curve segment extraction as discussed

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 520–527, 2012.

in the paper. In such an application, the class of hypersurface is parameterized linearly, and NLPCA works very well. But unfortunatelly, NLPCA sometimes derives an unacceptable estimation. The main reason for this is that errors of data are measured by the metric in the feature space, not in the input spafce (the space of observed data) in NLPCA. Since nonlinear mapping does not preserve distances, small error in the input space sometimes becomes large error in the feature space and vice versa. Then measuring errors in the feature space is not the best strategy in extracting structure of data. From this point of view, many methods considering the metric in the input space have been proposed [1,2,6,7,10,11]. These methods approximate feature mapping as affine mapping around each of the data by using *Jacobian matrix*, and then compute the relation between the metric in input space and that in feature space. By this approximation, the L_2 distance in the input space can be approximated as weighted L_2 distance by the quantities in the feature space and then it can be treated in the feature space. Then hypersurface fitting with the L_2 distance in the input space can be approximated by hyperplane fitting with the weighted L_2 distance in the feature space. The method is called *Jacobian NLPCA (JNLPCA)* [6].

On the other hand, detecting lines and quadratic curves in a two-dimensional image is an important and one of the most fundamental problems in image processing. To detect them, the *Hough transformation* (*HT*) [3], the *randomized Hough transformation* (*RHT*) [12], and *random sampling consensus* (*RANSAC*) [4] are frequently used. Since the HT and RHT estimates the parameters of lines or curves by voting on the cells in the parameter space, they have disadvantage that the accuracy of detected lines depends on the resolution of cells in the parameter space. The RANSAC estimates the parameters of lines or curves by counting the number of inliers, from which the distance to the lines or curves is less than the given threshold. Hence then the accuracy of detected lines depends on this threshold.

In this paper, two hypersurface fitting methods based on random sampling like RANSAC, are proposed. These methods use the L_2 distance in input space, not the distance in the feature space. The one method is *least α-percentile of squares* [5] ($L_\alpha PS$), which is an extension of *least median of squares* (*LMedS*) estimation [9]. The other is *least k-th power deviation* [5] ($L_k PD$), which is an extension of LS estimation. Briefly speaking, L_αPS uses the α-percentile instead of the median in LMedS, and L_kPD minimizes the sum of the k-th power deviations of errors instead of the sum of squares of errors. A remarkable property holds in one-dimensonal reduction by L_kPD for $0 < k \leq 1$, which is called *optimal sampling property*. The property is that there exists at least one global optimum which passes through N data points when an $N-1$ dimensional hyperplane is fitted to the N-dimensional data. By usig the property, finding optimum is reduced to combinatorial optimization of polynomial order, and it can be approximated by random sampling. The proposed methods are applied for fitting quadratic curve in an image, and detecting quadratic curve and/or ellipse segments.

2 Optimal Sampling Property and Jacobian NLPCA

For hyperplane fitting based on L$_k$PD, the following theorem holds:

Theorem 1.(Fujiki et al.[5]) Let $N-1$-dimensional affine space fitting for N-dimensional data points so as to minimize the (weighted) sum of k-th power L_p distance. When $0 < p \le \infty$ and $0 < k \le 1$, there exists a global optimum which passes at least N data points. (In the case of linear space fitting, a global optimum passes the origin and $N-1$ data points.) This property is called *optimal sampling property.*

When considering hypersurface fitting for the data in M-dimensional input space, the data are mapped into N-dimensional Hilbert space, which is called feature space, so as to have linear structure. By this mapping, hypersurface fitting is reduced to $N-1$-dimensional linear subspace fitting in the feature space. In usual NLPCA, the linear subspace is estimated by minimizing the sum of squared Euclidean distance between data and linear space in the feature space. Therefore, the approximation of the LS estimator in the input space is proposed [1,2,6,7,10,11].

The input space is assumed to be an M-dimensional Riemannian space, and its metric at observed data point $\boldsymbol{x}_{[d]}$ $(d = 1, \ldots, D)$ is denoted by $G_{[d]}$. In hypersurface fitting, each data $\boldsymbol{x}_{[d]}$ is mapped to the N-dimensional Hilbert space, called feature space, by the feature mapping $\boldsymbol{\phi} : \boldsymbol{x} \mapsto \boldsymbol{\phi}(\boldsymbol{x})$. By using $J_{[d]}$, which is the Jacobian matrix at $\boldsymbol{x}_{[d]}$, the metric in the feature space around $\boldsymbol{\phi}_{[d]} = \boldsymbol{\phi}(\boldsymbol{x}_{[d]})$ is linearly approximated by the metric in the input space as $\mathcal{G}_{[d]} = (J^{+}_{[d]})^{\top} G_{[d]} J_{[d]}$ where X^{+} is the Moore-Penrose inverse matrix of X. This paper considers that the set of fitting hypersurfaces is represented by a linear parameter $\boldsymbol{a}$ as $f(\boldsymbol{x}; \boldsymbol{a}) = \boldsymbol{a}^{\top}\boldsymbol{\phi}(\boldsymbol{x}) = 0$. For quadratic curve fitting in xy-plane, it is represented as $\boldsymbol{a}^{\top}(x^2, xy, y^2, x, y, 1)^{\top} = 0$. When the mapping $\boldsymbol{x} \mapsto \boldsymbol{\phi}$ is considered as a feature mapping, the $M-1$-dimensional hypersurface fitting problem is reduced to $N-1$-dimensional linear subspace fitting on feature space. In usual NLPCA, the distance between two points in the feature space is measured by L_2 distance in the feature space, but in JNLPCA, the distance between a point and a hypersurface in the feature space is measured by an approximation of the L_2 distance in the input space, and its representation is

$$R_{[d]} = \sqrt{\frac{\boldsymbol{a}^{\top}\left[\boldsymbol{\phi}_{[d]}\boldsymbol{\phi}_{[d]}^{\top}\right]\boldsymbol{a}}{\boldsymbol{a}^{\top}\mathcal{G}^{+}_{[d]}\boldsymbol{a}}}$$

as shown in Fujiki et al. [6]. Compared with this, the distance between the data and fitting hypersurface without considering the metric of input space is

$$r_{[d]} = \sqrt{\frac{\boldsymbol{a}^{\top}\left[\boldsymbol{\phi}_{[d]}\boldsymbol{\phi}_{[d]}^{\top}\right]\boldsymbol{a}}{\boldsymbol{a}^{\top}\boldsymbol{a}}}\,.$$

Then, the distance considering the metric of input space is represented by the weighted distance as

$$R_{[d]} = w_{[d]} r_{[d]}, \quad \text{where} \quad w_{[d]} = \sqrt{\frac{\boldsymbol{a}^\top \boldsymbol{a}}{\boldsymbol{a}^\top \mathcal{G}_{[d]}^{+} \boldsymbol{a}}}\,.$$

3 Random Sampling Approximation

Because hypersurface fitting is approximated by weighted hyperplane fitting in the feature space as discussed in the previous section, hyperplane fitting methods can be applied to hypersurface fitting. Then, in this paper, two hyperplane fittng methods, which are combinatorial optimization of polynomial order, are considered. The one is random sampling approximation of L$_k$PD$(0 < k \leq 1)$, which has the optimal sampling property, and the other is L$_\alpha$PS , which is an extension of LMedS. Since these methods are combinatorial optimization of polynomial order, finding the optimum takes high computational costs, even for quadratic curve fitting in $\mathbb{R}^2$ (For 100 data points, there are about 10^{12} combination to find the optimum.). To reduce the computational costs, the random sampling technique can be adopted. There exists a very famous random sampling method, which is known as RANSAC [4]. Roughly speaking, the difference among RANSAC, L$_\alpha$PS , and L$_k$PD are how to detect outliers. RANSAC classifies a data point into inlier when the approximated distance between the data point and the hypersurface is less than the given threshold e, and classifies the point into outlier otherwise, then finds the hypersurface which has the maximum number of inliers. L$_\alpha$PS defines the percentage of inliers by given α, and find the hypersurface so as to minimize the largest distance between inlier point and hypersurface (minimax criterion). L$_k$PD $(0 < k \leq 1)$ does not classify data as inlier or outlier, but effect of error of each data is rapidly decreasing when the error gets large. Note that when e, α, and k are getting smaller, these methods are getting more robust.

4 Experiments

Though fitting based on L_1 distance is more robust than that based on L_2 distance, the estimation by L_1 distance sometimes derives unacceptable result because of *leverage point* [9]. Generally, fitting based on L_p distance is getting more robust when p is getting small. Then, to reduce leverage point effect, 0.5-L$_k$PD is applied, for example. Figure 1 is a quadratic curve fitting results for artificial data. Data points are generated from an arc of parabola $y = x^2 (x \in [-3, 3])$ with uniform distribution and each point is contaminated by noise to normal direction of the parabola. The noise follows a Laplace distribution with 0-mean and 0.18-variance. The generated points are classified as inlier when the approximated distance between point and parabola is less than 0.3, and classified as outlier otherwise. 200 inliers and 50 outliers are generated. The fitting quadratic curve is

$$a_1x^2 + 2a_2xy + a_3y^2 + 2a_4x + 2a_5y + a_6 = 0\,.$$

In Fig. 1, the red solid curve shows fitting result, the green dotted curve shows the result by LS estimation in the feature space, and the blue dotted curve shows the result by LS estimation in the input space (2-L$_k$PD). From the top row of Fig. 1,

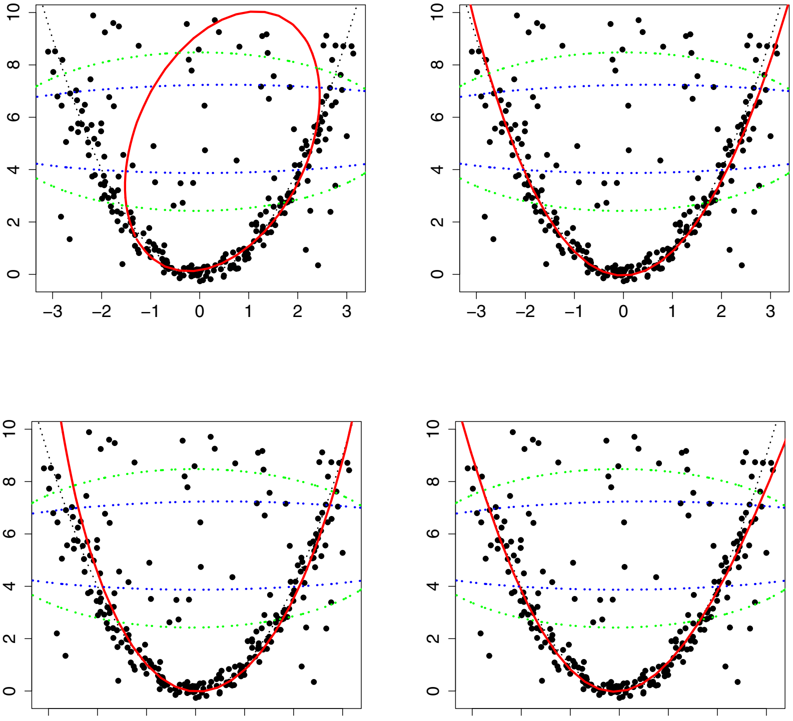


Fig. 1. Quadratic curve fittng: 1-L$_{\boldsymbol{k}}$PD (top left), 0.5-L$_{\boldsymbol{k}}$PD (top right), 0.25-L$_{\boldsymbol{\alpha}}$PS (bottom left), RANSAC (bottom right)

0.5-L$_k$PD reduces the effect by leverage point more than 1-L$_k$PD. From this experiment, it is shown that making k smaller is effective for robust estimation. These results are compared with the results by 0.25-L$_\alpha$PS and RANSAC. From Fig. 1, each of 0.5-L$_k$PD, 0.25-L$_\alpha$PS and RANSAC gives almost the same result when only one quadratic curve is fitted.

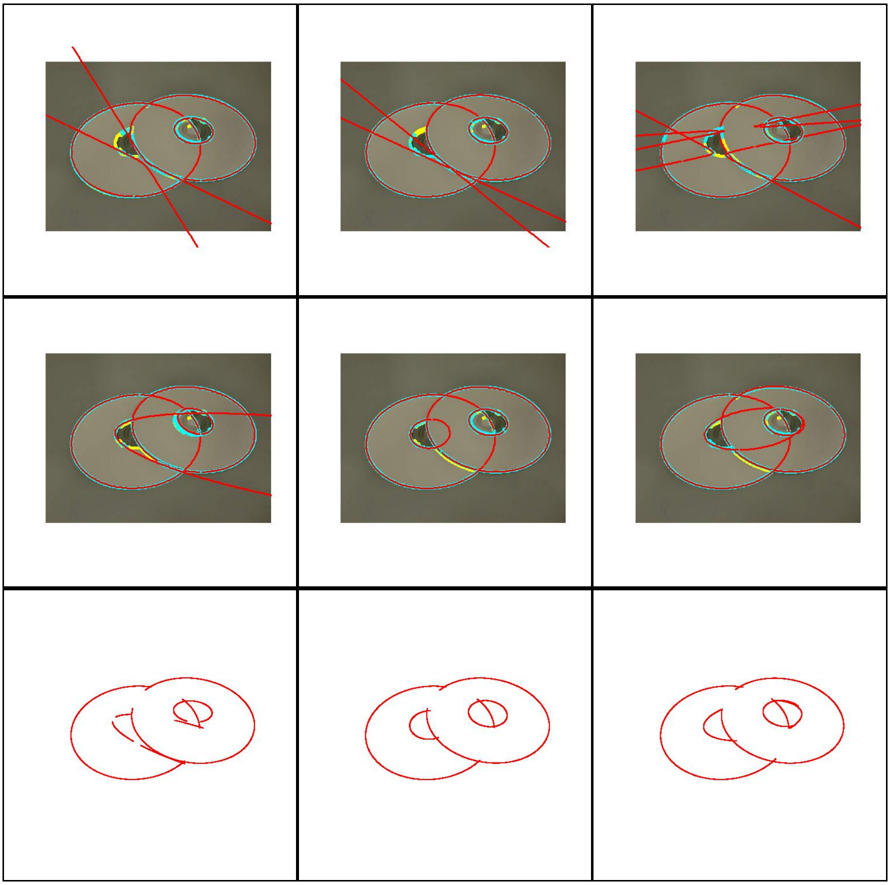


Fig. 2. Quadratic curve segments extractions: RANSAC (left), 20-L$_\alpha$PS (middle) and 0.05-L$_k$PD (right)

The proposed methods and RANSAC are applied to real-image data. The original 640 × 480 image is converted to an edge image by Canny filter with Gaussian convolution function of $\sigma = 2.0$, and 2918 points having the peak value in the edge image are chosen. Quadratic curve segment is assumed to consist of at least 20 points. When one curve segment is estimated, the points from estimated curve within $2\sqrt{5}$-pixel in approximated distance are regarded as inlier and removed. Then the same procedure is applied to the rest of the observed points till no curve segment is estimated. In the procedure, the number of random trials is determined as follows: When there are n points, the number of random trials such that at least one curve is passing through five inliers among 20 inliers in probability $1 - 10^{-4}$. Figure 2 shows the curves consist of more than 180 inliers. The top of Fig.2 shows the results of extracting quadratic curve segments by RANSAC, 20-L$_\alpha$PS and 0.05-L$_k$PD. The middle row of Fig.2 shows the results of extracting ellipses. In the extraction of ellipses, fitting quadratic curve is regarded as ellipse and accepted when $a_1a_3 - a_2^2 > 0$, and rejected otherwise. The bottom row of Fig.2 shows the extracted ellipse segments. To extract each segment, the density function of data on the segment is estimated by using function "density" in statisical soft R [8], and thresholding. As argued in previous section, RANSAC, L$_k$PD, and L$_\alpha$PS are similar in that they are developed to reduce the effect of outliers. From an experimental result with a real-image, most of curve segments in the picture are detected by either methods. As seen from Fig.2 (middle), L$_\alpha$PS is suitable for line segments estimation and it can be used instead of RANSAC or HT. It is noted that, in this experiment, L$_k$PD does not give the best performance. The reason why the result of L$_k$PD is slightly inferior to others is that L$_k$PD considers the effect of all points, while RANSAC and L$_\alpha$PS only consider the effect of the points around line segments.

5 Conclusion

This paper proposed hypersurface fitting methods based on random sampling. The methods use the relation between the metric in the input space and that in the feature space. The proposed methods are evaluated by quadratic curve fitting. From the experiments, L$_\alpha$PS is competitive with RANSAC in extraction of hypersurfaces. But L$_k$PD is not competitive with RANSAC in extraction multiple hypersurfaces. The reason why the result of L$_k$PD is slightly inferior to others is that L$_k$PD considers the effect of all points, while RANSAC and L$_\alpha$PS only consider the effect of the points around curve segments. This consideration derives that L$_k$PD is suitable for extracting only one structure embedding in data. In quadratic curve extraction, the phenomena which does not occur in line segments extraction. This is that some hyperbola fits to a common tangent of two quadratic curves as you can see in Fig. 2. To avoid the phenomena, the density of observed data point should be investigated.

References

1. Akaho, S.: Curve fitting that minimizes the mean square of perpendicular distances from sample points. SPIE, Vision Geometry II (1993)
2. Chojnacki, W., Brooks, M.J., van den Hangel, A., Gawley, D.: On the fitting of surface to data with covariances. IEEE TPAMI 22(11) (2000)
3. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Comm. ACM 15, 11–15 (1972)
4. Fischer, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. ACM 24, 381–395 (1981)
5. Fujiki, J., Akaho, S., Hino, H., Murata, N.: Robust Hyperplane Fitting Based on k-th Power Deviation and α-Quantile. In: Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (eds.) CAIP 2011, Part I. LNCS, vol. 6854, pp. 278–285. Springer, Heidelberg (2011)

6. Fujiki, J., Akaho, S.: Hypersurface Fitting via Jacobian Nonlinear PCA on Riemannian Space. In: Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (eds.) CAIP 2011, Part I. LNCS, vol. 6854, pp. 236–243. Springer, Heidelberg (2011)
7. Kanatani, K., Sugaya, Y.: Unified computation of strict maximum likelihood for geometric fitting. Journal of Mathematical Imaging and Vision 38(1), 1–13 (2010)
8. R Development Core Team, R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2008), http://www.R-project.org ISBN3-900051-07-0
9. Rousseeuw, R.J., Leroy, A.M.: Robust Regression and Outlier Detection. John Wiley & Sons (1987)
10. Sampson, P.D.: Fitting conic sections to 'very scattered' data: an iterative refinement of the Bookstein algorithm. Comput. Vision, Graphics, and Image Processing 18, 97–108 (1982)
11. Taubin, G.: Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applicatons to edge and range image segmentation. IEEE TPAMI 13(11) (1991)
12. Xu, L., Oja, E., Kultanan, P.: A new curve detection method: Randomized Hough Transform (RHT). Pattern Recognition Letters 11, 331–338 (1990)

Manifold Regularized Multi-Task Learning

Peipei Yang, Xu-Yao Zhang, Kaizhu Huang, and Cheng-Lin Liu

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences,
Beijing, China 100190
{ppyang,xyz,kzhuang,liucl}@nlpr.ia.ac.cn

Abstract. Multi-task learning (MTL) has drawn a lot of attentions in machine learning. By training multiple tasks simultaneously, information can be better shared across tasks. This leads to significant performance improvement in many problems. However, most existing methods assume that all tasks are related or their relationship follows a simple and specified structure. In this paper, we propose a novel manifold regularized framework for multi-task learning. Instead of assuming simple relationship among tasks, we propose to learn task decision functions as well as a manifold structure from data simultaneously. As manifold could be arbitrarily complex, we show that our proposed framework can contain many recent MTL models, e.g. RegMTL and cCMTL, as special cases. The framework can be solved by alternatively learning all tasks and the manifold structure. In particular, learning all tasks with the manifold regularization can be solved as a single-task learning problem, while the manifold structure can be obtained by successive Bregman projection on a convex feasible set. On both synthetic and real datasets, we show that our method can outperform the other competitive methods.

Keywords: Multi-task Learning, Manifold Learning, Laplacian.

1 Introduction

In many machine learning problems, we usually have multiple corrected learning problems or tasks. Traditionally we can train each task from its training samples individually. However, if the number of training samples in each task is small, they tend to be overfitting, meaning that the performance is very likely to be bad for future samples. To handle this problem, multi-task learning (MTL) manages to learn all tasks simultaneously. By sharing information across related tasks, MTL can usually lead to better performance than the traditional single task learning.

However, in order to share information appropriately, MTL often needs to assume how the tasks are correlated. Given that a linear decision function is to be learned for each task, the relationship among tasks can be specified directly via the weight vectors associated with the decision function. For example, [7] proposed the *Regularized Multi-task Learning* (RegMTL) which assumes that the weight vector of each task is composed with a common part and an individual

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 528–536, 2012.

part. The common part contains the shared information of all the tasks and the propagation of information is enforced by minimizing the individual part for each task. It equivalently implies that the weight vectors of different tasks belong to a ball of an unknown center determined by the common part.

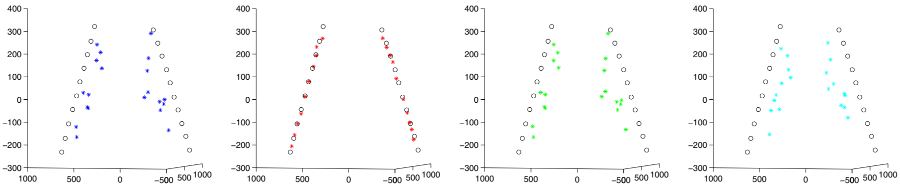
Unfortunately, such an assumption may be too strict in practice, since it is unnecessary for each task to be related with all other tasks. To solve this problem, [9] generalized this assumption to the case that these tasks can gather into several clusters and proposed the *convex Clustered Multi-task Learning* (cCMTL) [10] method. Within each cluster, it is a traditional MTL problem, i.e., the weight vectors of different tasks in a certain cluster are in a ball of an unknown center determined by the common part. cCMTL can learn the weight vectors of all tasks and the cluster structure simultaneously.

Although cCMTL provides a tool to capture the topological structure of the relationship among tasks, its assumption is still too strong and may be too simple to explore the actual task relationship. On one hand, tasks may be unable to be partitioned into several groups. On the other hand, even if several tasks belongs to a cluster, it never means each task within this cluster is correlated with each other at the same level.

Hence, the structure of the relationship between tasks could be more complex, and a general manifold structure should be considered. Take an example for illustration. Consider the problem that there are 20 related regression tasks. To show the relationship between tasks, we plot the weight vectors in Fig. 1 using hollow points in a 3-dimensional space. From this figure, the weight vectors gather into 2 clusters and each of them forms a 1-dimensional manifold. To the extent of our knowledge, there has not been a method designed to deal with this case.

Since manifold has the ability to describe not only the topological structure of data, but also the local metric structure, we propose Manifold Regularized Multitask Learning (MRMTL) which engages manifold to capture the relationship among the tasks. All tasks and the manifold structure of their relationship are learned simultaneously, and both of them are improved with the help of each other. As manifold could be arbitrarily complex, we show that our proposed framework can contain many recent MTL models, e.g. RegMTL and cCMTL, as special cases. Moreover, the proposed framework can be solved by learning all tasks and the manifold structure alternatively. In particular, learning all tasks with the manifold regularization can be solved as a single-task learning problem, while the manifold structure can be obtained by successive Bregman projection on a convex feasible set. It is noticeable that [8] has studied the multitask learning problem with manifold regularization. However, it supposed that the manifold structure is given preliminarily. As a key difference, our proposed approach can learn the manifold from the training samples automatically.

The rest part of this paper is organized as follows. In Section 2, we first present the problem definition and then introduce the basic framework of our method. In Section 3, the optimization algorithm is given in detail. In Section 4, we evaluate our method on a synthetic dataset and a real dataset, both of which show the effectiveness of our method. At last, we set out the final remarks in Section 5.



(a) STL (b) MRMTL (c) cCMTL (d) RegMTL

Fig. 1. Learned weight vectors W using different methods. The hollow points are ground truth while the star points are learned results.

2 Problem Definition and Main Framework

In this section, we first present the notation and problem definition. We then introduce the framework of Manifold Regularized Multi-task Learning in detail.

2.1 Notation and Problem Definition

In this paper, we consider the problem where a linear decision function is learned and thus the aim of each task is to learn a weight vector. Suppose there are n tasks. For the t-th task, we have a training data set $\mathcal{X}_t$ containing m_t data points $\mathbf{x}_{tk} \in \mathbb{R}^d$ whose dimension is d and a corresponding output set $\mathcal{Y}_t$ containing the target output y_{tk}. For binary classification problem, $\mathcal{Y}_t = \{-1, +1\}$, while for regression problem, $\mathcal{Y}_t = \mathbb{R}$.

We use $l\,(y, f(\mathbf{x}))$ to quantify the loss of predicting $f(\mathbf{x})$ for the input $\mathbf{x}$ when the expected output is y, which depends on the problem. For example, in binary classification, the hinge loss $l\,(y, f(\mathbf{x})) = \max(0, 1 - y \cdot f(\mathbf{x}))$ is often used, while in regression, the squared error $l\,(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$ is often chosen. If the linear prediction function $f(\mathbf{x}) = \mathbf{w}_t^\top \mathbf{x}$ is used and we denote $W = [\mathbf{w}_1\ \mathbf{w}_2\ \ldots\ \mathbf{w}_n]$, the empirical loss of all tasks can be then formulated as $\ell(W) = \sum_{t=1}^{n} \sum_{j=1}^{m_t} l(y_{tj}, \mathbf{w}_t^\top \mathbf{x}_{tj})$.

2.2 Coupling Multiple Tasks with Regularization

In order to learn all the tasks simultaneously, we follow the well-established method that embeds the relationship among tasks into a regularization item and use a graph to describe the relationship among tasks. Specifically, each vertex of the graph represents a task, and each edge linking two vertices indicates the relationship between the two tasks. A greater weight of edge represents a closer relationship. Define S as the weight matrix of this graph where S_{ij} is the weight of the edge connecting the i-th and j-th vertices and D is a diagonal weight matrix whose entries are column sums of S, then $L = D - S$ is the Laplacian matrix [5] of this graph.

In Laplacian regularization [2], we have $\mathrm{tr}(WLW^\top) = \sum_{i,j} \frac{1}{2} \|\mathbf{w}_i - \mathbf{w}_j\|^2 S_{ij}$, which can be then used as the regularization to enforce the linked pairs to be more similar. If the i-th and j-th tasks are closely correlated, the corresponding edge weight S_{ij} is large, which encourages $\|\mathbf{w}_i - \mathbf{w}_j\|^2$ to be less and thus the learned weight vectors $\mathbf{w}_i$ and $\mathbf{w}_j$ are more liable to be similar.

However, in MTL, such task similarity S_{ij} is unavailable beforehand and should be learned from data. It is obvious that if we directly optimize on L and W simultaneously, we will simply obtain the Laplacian matrix L with all elements zero regardless of W. Therefore, in order to discover the relationship among tasks, we should add some additional constraints on L. Without more prior knowledge, a Laplacian matrix of a graph whose vertices are all connected may be a reasonable prior of L. Therefore, we get the following optimization formula of MRMTL

$$\min_{W,L} \mathcal{R}(W,L) = \sum_{t=1}^{n}\left(C\sum_{j=1}^{m_t} l(y_{tj}, \mathbf{w}_t^\top \mathbf{x}_{tj}) + \mathbf{w}_t^\top \mathbf{w}_t\right) + \gamma\left(\mathrm{tr}(WLW^\top) + \frac{\gamma_0}{2}\|L - L_0\|_{\mathrm{F}}^2\right)$$
$$\text{s.t. } L\mathbf{1}_n = \mathbf{0}; \quad L = L^\top; \quad L_{ij} \le 0, \forall i \ne j$$

where L_0 is the Laplacian matrix for a graph with all nodes connected ($S_{ij} = 1, \forall i,j$), i.e., $L_0 = n(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)$, where $W = [\mathbf{w}_1\ \mathbf{w}_2\ \ldots\ \mathbf{w}_n]$, and $\mathbf{1}_n$ is an n-dimensional vector with all elements 1. In this formulation, l is the loss from training samples and $\mathbf{w}_t^\top \mathbf{w}_t$ is the regularization. Both of them are determined by the original learning problem. $\mathrm{tr}(WLW^\top)$ is the manifold regularization which enforces the weight vectors of similar tasks to be similar. The last term provides a prior for L and prevents the trivial solution for L. The constraints guarantee that L is a Laplacian matrix, which is therefore also symmetric positive semi-definite.

2.3 Relationship with Other Methods

It is noticeable that our method includes RegMTL as a special case. Indeed, if we choose γ_0 to be large enough, we will get $L = L_0$ and the regularization item becomes $\mathrm{tr}(WL_0W^\top) = n \cdot \left\|W - \bar{\mathbf{w}}\mathbf{1}_n^\top\right\|_{\mathrm{F}}^2 = n\sum_{t=1}^{n}\|\mathbf{w}_t - \bar{\mathbf{w}}\|^2$.

By Lemma 2.2 of [7], this problem is an alternative formulation of RegMTL and thus it is just a special case of MRMTL. We can also regard MRMTL as a generalized of RegMTL in which the relationship among tasks is learned using L_0 as prior, rather than to use L_0 directly.

When the tasks gather into several clusters, [9] uses the $m \times r$ binary matrix E to denote the cluster assignment where $E_{ij} = 1$ if task-i belongs to cluster-j and $E_{ij} = 0$ otherwise. Define $M = E(E^\top E)^{-1}E^\top$, $U = \mathbf{1}_m\mathbf{1}_m^\top/m$, then M is the edge weight matrix of the graph of tasks where $M_{ij} = 1/m_c$ if task-i and task-j belong to the same cluster-c and $M_{ij} = 0$ otherwise, where m_c is the number of tasks in cluster-c. The regularization with respect to the task clustering is

$$\mathrm{tr}(WKW^\top) = \mathrm{tr}\left(W\left(\varepsilon_B(M - U) + \varepsilon_W(\mathbf{I} - M)\right)W^\top\right).$$

It is easy to verify that K is a Laplacian matrix if $\varepsilon_W \ge \varepsilon_B$, which is satisfied in cCMTL [9]. Therefore, our method indeed also includes clustered multi-task learning as a special case in the sense that any cluster structure of the tasks can be formulated using our model. However, the solution may be different since our model is more flexible to fit the data.

3 Optimization

In this section, we first present how to solve the problem using alternative optimization, and then show how each step of the optimization is solved.

3.1 Alternative Optimization

This problem can be solved by alternative optimization. Specifically, we solve for an optimal $W^{(1)}$ with $L = L^{(0)}$ fixed as an initial value first, and then solve for an optimal $L^{(1)}$ with $W = W^{(1)}$ fixed. Such procedure is then repeated so that both L and W are optimized alternatively until convergence. Since l is usually chosen to have a lower bound and L is constrained to be positive semi-definite, there exists a lower bound for $\mathcal{R}(W, L)$. In another respect, the value of objective function decreases in each iteration, and thus it is guaranteed to converge to a local minimal value after certain iterations.

Note that the optimization is not convex and the global optimal solution is not guaranteed. Nevertheless, we found that given a proper initial solution, the local optimal solution is often good enough. Since W is solved firstly, we should specify an initial point for L. An appropriate choice is $L^{(0)} = L_0$. With this choice, $W^{(1)}$ is indeed the solution of RegMTL. After several iterations, the incorrect connections in the graph are removed and the manifold can be eventually learned.

In the following of this section, we will give the algorithm to solve W and L respectively in detail.

3.2 Fix L and Optimize on W

The part of $\mathcal{R}$ with respect to W is

$$\mathcal{R}(W) = \sum_{t,j} C \cdot l(y_{tj}, \mathbf{w}_t^\top \mathbf{x}_{tj}) + J(W), \text{ where } J(W) = \operatorname{tr}\left(W(\mathbf{I}_n + \gamma L)W^\top\right) \quad (1)$$

Denote $\mathbf{w} = \operatorname{vec}(W) = \left[\mathbf{w}_1^\top \ \mathbf{w}_2^\top \ \ldots \ \mathbf{w}_n^\top\right]^\top$ as the vector concatenated by $\{\mathbf{w}_t\}$, then by Proposition 31 of [3], we have $\operatorname{vec}(Y)^\top (A \otimes B)\operatorname{vec}(X) = \operatorname{tr}(A^\top Y^\top BX)$ and thus

$$J(W) = \operatorname{tr}((\mathbf{I}_n + \gamma L)W^\top \mathbf{I}_d W) = \mathbf{w}^\top E \mathbf{w} = J(\mathbf{w}), \text{ where } E = (\mathbf{I}_n + \gamma L)^\top \otimes \mathbf{I}_d.$$

Suppose $B^\top B = E^{-1} = ((\mathbf{I}_n + \gamma L)^\top \otimes \mathbf{I}_d)^{-1} = (\mathbf{I}_n + \gamma L)^{-1} \otimes \mathbf{I}_d$ and consider the problem

$$\min_{\mathbf{u}} \mathcal{S}(\mathbf{u}) = \sum_t \sum_j C \cdot l(y_{tj}, \mathbf{u}^\top B_t \mathbf{x}_{tj}) + \mathbf{u}^\top \mathbf{u}, \text{ where } B = [B_1 \ B_2 \ \ldots \ B_n]. \quad (2)$$

By Proposition 1 of [8], we have $\mathcal{S}(\mathbf{u}) = \mathcal{R}(B^\top \mathbf{u})$. Thus the optimal solution of (1) can be obtained by solving the single-task problem (2) and $\mathbf{w}_t = B_t^\top \mathbf{u}$.

3.3 Fix W and Optimize on L

When W is fixed, the optimization problem on L becomes

$$\begin{aligned} &\min_L \mathcal{R}(L) = \gamma \left(\frac{\gamma_0}{2} \|L - L_*\|_{\mathrm{F}}^2 + \mathcal{R}_{\mathrm{const}} \right) \\ &\text{s.t. } L\mathbf{1}_n = \mathbf{0}; \quad L = L^\top; \quad L_{ij} \leq 0, \forall i \neq j \end{aligned} \tag{3}$$

where $L_* = L_0 - \frac{1}{\gamma_0} W^\top W$ and $\mathcal{R}_{\mathrm{const}}$ is a constant independent of L. This is a *Bregman projection* problem [6] whose optimal solution is the projection of L_* on the convex set $\mathbf{C}_1 \cap \mathbf{C}_2$ where $\mathbf{C}_1 = \{L \in \mathbb{R}^{n \times n} \mid L\mathbf{1}_n = \mathbf{0}; L = L^\top\}$ and $\mathbf{C}_2 = \{L \in \mathbb{R}^{n \times n} \mid L_{ij} \leq 0, \forall i \neq j\}$. The optimal solution of L can be obtained by Successive Projection-Correction Algorithm (Algorithm B of [6]) on these two convex sets.

Projection onto $\mathbf{C_1}$. The Lagrangian formulation[1] of the projection on $\mathbf{C}_1$ is

$$\min_{L, \mu_1, \mu_2} \|L - L_*\|_{\mathrm{F}}^2 - \mu_1^\top L\mathbf{1}_n - \mu_2^\top L^\top \mathbf{1}_n$$

where from the condition $L = L^\top$ we have $\mu_1 = \mu_2 = \mu$. Setting the derivative with respect to L to zero yields $L = L_* + \frac{1}{\gamma_0}\mu\mathbf{1}_n^\top + \frac{1}{\gamma_0}\mathbf{1}_n\mu^\top$. Multiplying with $\mathbf{1}_n$ on the right of both sides of the equation, then using Sherman-Morrison inverse formula [1] and $L_* = L_*^\top$, we have

$$\mu = -\gamma_0 \left(n\mathbf{I}_n + \mathbf{1}_n\mathbf{1}_n^\top\right)^{-1} L_*\mathbf{1}_n = \frac{\gamma_0}{n}\left(\frac{1}{2n}\mathbf{1}_n\mathbf{1}_n^\top - \mathbf{I}_n\right) L_*\mathbf{1}_n$$

Then substituting it into the formula of L, we get

$$L = L_* + \frac{1}{n^2}\left(\mathbf{1}_n^\top L_*\mathbf{1}_n\right)\mathbf{1}_n\mathbf{1}_n^\top - \frac{1}{n}\left(L_*\mathbf{1}_n\mathbf{1}_n^\top + \mathbf{1}_n\mathbf{1}_n^\top L_*\right)$$

Projection onto $\mathbf{C_2}$. The projection onto $\mathbf{C}_2$ can be obtained by simply setting the positive non-diagonal elements to zero following a correction step [6].

4 Experiments

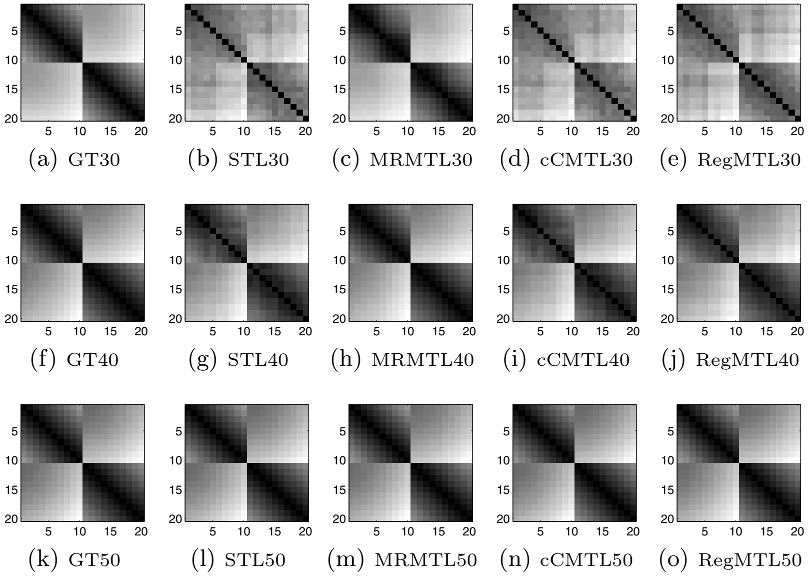
In this section, we empirically evaluate our method on both artificial data and real data. We apply our method on regression problems, and the normalized mean square error (nMSE) [4] is used as the performance measure. Specifically, it is defined as the mean squared error (MSE) divided by the variance of the target vector.

[1] The coefficient $\frac{\gamma\gamma_0}{2}$ is simply omitted.

We compare our method MRMTL with cCMTL [9], RegMTL [7], and single-task (STL) method as baseline. For each method, we use 5-fold cross validation to determine the regularization parameters.

4.1 Synthetic Data

We first evaluate on synthetic data set to give a visualized comparison of the results learned by these methods. We generate 20 related regression tasks using 20 weight vectors and then generate a certain number of training samples and 500 testing samples. The weight vectors are learned with the training samples using different methods and tested with the testing samples. We show the learned task relationship in Fig. 2 which is a 20 × 20 grid. The color of the grid on row-i and column-j represents the squared Euclidean distance of $\mathbf{w}_i$ and $\mathbf{w}_j$. From the results, we see that MRMTL can learn the task relationship surprisingly well, which coincides with the ground truth perfectly when the training samples is equal to 30, 40, and 50. It always gives the best performance compared with the other methods. Particularly, it demonstrates a significantly better performance than the other methods when the training samples are fewer. For the case where the number of training samples per task is 30, we also show the learned weight vectors in the 3-dimensional principal component subspace in Fig. 1. The hollow points represents



(a) GT30 (b) STL30 (c) MRMTL30 (d) cCMTL30 (e) RegMTL30

(f) GT40 (g) STL40 (h) MRMTL40 (i) cCMTL40 (j) RegMTL40

(k) GT50 (l) STL50 (m) MRMTL50 (n) cCMTL50 (o) RegMTL50

Fig. 2. Comparison of the weight vectors learned by different methods. The five columns of this figure correspond to the (1)Ground Truth (GT); (2)Single-task Learning (STL); (3)Manifold Regularization Multi-task Learning (MRMTL); (4)convex Clustered Multi-task Learning (cCMTL); (5)Regularized Multi-task Learning (RegMTL). The number in the title indicates how many percent of training samples are used.

the ground truth while star points represent the learned results. We see again that MRMTL gives the best result and the manifold is learned exactly.

4.2 Real Data

We also evaluate these methods on Sarcos data[2] [10], which relates to an inverse dynamics prediction problem for a seven degrees-of-freedom anthropomorphic robot arm. It consists of 48933 observations corresponding to 7 joint torques; each of the observations is described by 21 features including 7 joint positions, 7 joint velocities, and 7 joint accelerations. The prediction of each joint torque corresponds to one task. We randomly select 10, 20, 50, 100 samples from each task for training and the remaining for test. The experiment is repeated 5 times and the averaged nMSE (the less the better) are shown in Table. 1. From the results, we can observe that MRMTL performs the best, regardless of the number of samples used for training.

Table 1. Performance comparison on Sarcos Dataset using nMSE

Sample	STL	MRMTL	cMTL	RegMTL
10	2.8788	1.7843	2.7532	2.8867
20	0.8383	0.5487	0.7953	0.5766
50	0.2615	0.1709	0.4377	0.2066
100	0.1664	0.1188	0.3378	0.1202

5 Conclusion

In this paper, we propose a novel manifold regularized framework for multi-task learning. Different from recent work that usually assumes simple relationship among tasks, we propose to learn task decision functions as well as a manifold structure from data simultaneously. We show that our proposed framework can subsume many recent MTL models, e.g. RegMTL and cCMTL, as special cases. Moreover, the framework can be solved by alternatively learning all tasks and the manifold structure. A series of experiments on both synthetic and real data show that our method can significantly outperform the other competitive methods.

Acknowledgements. This work has been supported in part by the National Basic Research Program of China (973 Program) Grant 2012CB316301, the National Natural Science Foundation of China (NSFC) Grants 61075052 and 60825301, and Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation.

[2] http://gaussianprocess.org/gpml/data/

References

1. Bartlett, M.S.: An Inverse Matrix Adjustment Arising in Discriminant Analysis. The Annals of Mathematical Statistics 22(1), 107–111 (1951)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373–1396 (2002)
3. Broxson, B.J.: The kronecker product. UNF Theses and Dissertations (2006)
4. Chen, J., Zhou, J., Ye, J.: Integrating low-rank and group-sparse structures for robust multi-task learning. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 42–50 (2011)
5. Chung, F.R.K.: Spectral Graph Theory (CBMS Regional Conference Series in Mathematics), vol. 92 American Mathematical Society (February 1997)
6. Dhillon, I.S., Tropp, J.A.: Matrix nearness problems with bregman divergences. SIAM Journal on Matrix Analysis and Applications 29, 1120–1146 (2008)
7. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 109–117 (2004)
8. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. Journal of Machine Learning Research 6, 615–637 (2005)
9. Jacob, L., Bach, F., Vert, J.P.: Clustered multi-task learning: A convex formulation. In: NIPS, pp. 745–752 (2008)
10. Zhou, J., Chen, J., Ye, J.: Clustered multi-task learning via alternating structure optimization. Advances in Neural Information Processing Systems 24, 702–710 (2011)

Cooperative Behavior Acquisition in Multi-agent Reinforcement Learning System Using Attention Degree

Kunikazu Kobayashi[1], Tadashi Kurano[2],
Takashi Kuremoto[2], and Masanao Obayashi[2]

[1] Aichi Prefectural University, 1522-3 Ibaragabasama, Nagakute, Aichi 480-1198, Japan
[2] Yamaguchi University, 2-16-1 Tokiwadai, Ube, Yamaguchi 755-8611, Japan
kobayashi@ist.aichi-pu.ac.jp,
{wu,m.obayas}@yamaguchi-u.ac.jp
http://www.ist.aichi-pu.ac.jp/~koba/

Abstract. In a multi-agent system, it becomes possible to solve a complicated problem by cooperative behavior with others. When people act in a group, as they are predicting the others' action, estimating the others' intention, and also making eye contact with others, they are realizing cooperative behavior efficiently. In the present paper, we try to introduce the concept of eye contact into a multi-agent system. In order to realize eye contact, we firstly define attention degrees both from self to the other and from the other to self. After that, we propose an action decision method that self agent makes easy to choose a target agent and to choose actions approaching to the agent using the attention degrees. Through computer simulation using a pursuit problem, we show that the agents making eye contact each other pursue preys by approaching each other. Simultaneously, we compare the proposed system with the standard Q-learning system and verify the usefulness of the proposed system.

Keywords: Multi-agent system, Cooperative behavior, Eye contact, Attention degree, Reinforcement learning, Pursuit problem.

1 Introduction

In multi-agent systems (MASs), intellectual behavior such as cooperative behavior can emerge toward a goal of agent group through mutual interaction among individual agents. In general, multi-agent systems have three major advantages over single-agent systems (SASs): robustness, flexibility, and load sharing [1]. As giving agents a reinforcement learning function, MASs can maximize its potential abilities such as problem solving and adaptation abilities [2,3].

To realize cooperative behavior in MASs, if agents are able to communicate with others using a highly accurate communication tool, agents can accurately obtain the other's action or intention [4]. Agents however has to predict the other's action or estimate the other's intention if agents are unable to communicate with others by restrictions of robot hardware and external environments.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 537–544, 2012.

Nagayuki et al. presented a policy estimation method which can estimate the other's action to be taken based on the observed information about the other's action sequence [5,6]. They successfully applied it to the reinforcement Q-learning method [7] and showed to get effective the other's policy. Meanwhile, Yokoyama et al. proposed an approach to model action decision based on the other's intention according to atypical situation such as human-machine interaction [8,9]. They presented three estimation levels of the other's intention and presented a computational model of action decision process to solve cooperative tasks through a psychological approach. In this context, Kobayashi et al. successfully presented an adaptive approach for automatically switching the above three estimation levels depending on the situation [10].

In the present paper, in order to realize cooperative behavior, we introduce a concept of eye contact, which is motivated by a method for detecting focusing intention of the learner in the collaborative learning environment [11]. Firstly, we formulate eye contact by a Q-value in reinforcement Q-learning method [7] and a P-value in the policy estimation method [5,6]. Secondly, we propose two kinds of attention degrees both from self to the other and from the other to self. Thirdly, we propose an action decision method that self agent makes easy to choose a target agent and to choose actions approaching to the agent using the attention degrees. Finally, through computer simulations using a pursuit problem, we show that the agents making eye contact each other pursue preys by approaching each other. Simultaneously, we compare the proposed system with the standard Q-learning system and verify the usefulness of the proposed system.

2 Reinforcement Learning

Reinforcement learning is a machine learning technique that a decision-making agent takes actions and then receives rewards in an environment, and finally acquires the optimum policy by trial and error [2,3].

The Q-learning method by Watkins et al. is a representative reinforcement learning technique and guarantees that a value function will converge to the optimal solution by appropriately adjusting a learning rate in Markov decision process environments [7]. A state-action value function $Q(s, a)$ (Q-value) is updated by (1) so as to take the optimal action by exploring it in a learning space.

$$Q(s, a) \leftarrow Q(s, a) + \alpha\delta, \tag{1}$$

where α is a learning rate ($0 < \alpha < 1$) and δ is a temporal difference error (TD error) denoted by (2).

$$\delta = r + \gamma \max_{b \in A} Q(s', b) - Q(s, a), \tag{2}$$

where r is a reward at the state s', s' is the next state after an agent takes action a, γ is a discount rate ($0 \leq \gamma \leq 1$), and A is a set of all possible actions.

Probabilistically, an agent selects action a at state s according to policy $\pi(s, a)$. Throughout the present paper, we employ the Boltzmann distribution defined by (3) as the policy.

$$\pi(s,a) = \frac{\exp(\beta Q(s,a))}{\sum_{b \in A} \exp(\beta Q(s,b))}, \tag{3}$$

where β is a parameter to control randomness of action selection called as inverse temperature parameter. The policy $\pi(s,a)$ refers to a probability to select action a at state s.

3 Multi-Agent Systems

MASs have three major advantages over SASs: robustness, flexibility, and load sharing [1]. In higher dimensional space, however, MASs have so-called state-space explosion problem. The tile coding to be describe in Section 3.1 is well-known for solving the above problem. On the other hand, to realize cooperative behavior, we focus on predicting other's action to be described in Section 3.2 and attention degree to be proposed in Section 4.

3.1 Tile Coding

First of all, in the present paper, we consider a grid world as an external environment. To overcome state-space explosion problem in MASs, we firstly consider a generalization of state-space by random tile-coding [3]. As shown in Fig. 1, the state-space is randomly covered by some square tiles in order to reduce the number of states.

By apply this random tile-coding to Q-learning, we get the following Q-value.

$$Q(s,a) = \sum_{i=1}^{n} q(i,a)\phi(i), \tag{4}$$

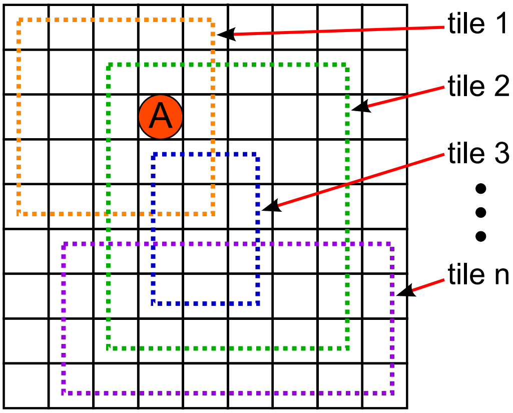


Fig. 1. Random tile coding (circle: agent, solid line: state space, dashed line: tiles)

where n denotes the number of tiles, $q(i, a)$ is the value function with respect to tile i and action a, and $\phi(i)$ is a binary function whether the agent exists in tile i or not as defined by:

$$\phi(i) = \begin{cases} 1 & \text{if the agent exists in tile } i, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The q-value is updated by (6), which is similar with the update rule of Q-value in (1):

$$q(s, a) \leftarrow q(s, a) + \alpha\delta/n. \tag{6}$$

3.2 Policy Estimation

The policy estimation method can estimate the other's action to be taken based on the observed information about the other's action sequence [5,6]. The method predicts an other's action using a policy estimation function $P(s, a_o)$ (P-value). The P-value is updated by (7) for all the other's actions to be taken $a_o \in A$

$$P(s, a_o) \leftarrow (1 - \rho)P(s, a_o) + \begin{cases} \rho & \text{if } a_o = a_o^*, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where a_o^* is the actual other's action and ρ is a positive parameter ($0 \leq \rho \leq 1$). As updating P-value by (7), P-value with a_o^* increases and the other P-values decrease. Repeatedly updating P-values, an agent can predict other's actions. It should be noted that the following relation holds at any time:

$$\sum_{a_o \in A} P(s, a_o) = 1. \tag{8}$$

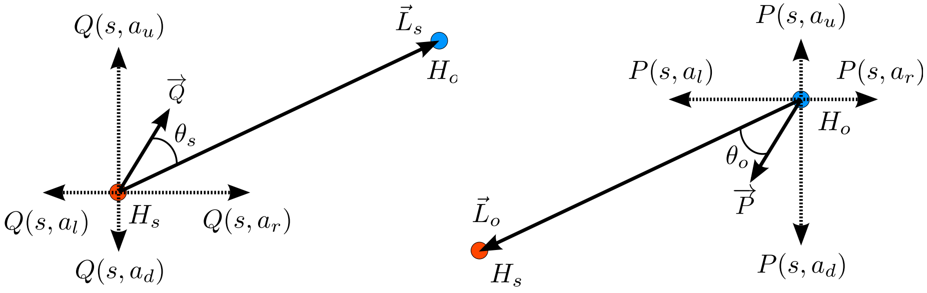
4 Intelligent Learning System Using Attention Degree

In order to emerge cooperative behavior, we introduce a concept of eye contact. It is motivated by a method for detecting focusing intention of the learner in the collaborative learning environment [11]. In the present paper, an intelligent learning system using attention degree is proposed. At first, we realize eye contact by attention degree (Section 4.1). Then, we propose an action decision method using attention degree (Section 4.2).

4.1 Attention Degree

First of all, we treat two kinds of attentions: attention from self to the other and attention from the other to self. Then, these two kinds of attentions are quantified by a Q-value in (1) and a P-value in (7). The attention degrees both from self to the other and from the other to self are illustrated in Fig.2. Note that both self agent H_s and the other agent H_o are placed on a grid world.

In Fig.2(a), we consider four Q-values at state s, i.e. $Q(s, a_u)$, $Q(s, a_d)$, $Q(s, a_l)$, and $Q(s, a_r)$. Here, a_u, a_d, a_l, and a_r refer to actions of moving up, down, left, and



(a) A resultant vector $\overrightarrow{Q}_s$ for H_s and a positional vector $\overrightarrow{L}_s$ for H_o.

(b) A resultant vector $\overrightarrow{P}_o$ for H_o and a positional vector $\overrightarrow{L}_o$ for H_s.

Fig. 2. Illustration of resultant and positional vectors

right, respectively. Then, we calculate a resultant vector $\overrightarrow{Q}$ and an angle θ_s between $\overrightarrow{Q}$ and a positional vector $\overrightarrow{L}_s$ for H_o. Finally, an attention degree from self to the other $AD(H_s, H_o)$ is defined by:

$$AD(H_s, H_o) = (\cos\theta_s + 1)/2. \tag{9}$$

It is clear that $AD(H_s, H_o)$ has value between 0 and 1, i.e. $AD(H_s, H_o) \in [0, 1]$. $AD(H_s, H_o)$ approaches to 1 as H_o pays attention to H_s.

Similarly, we consider four P-values at state s, i.e. $P(s, a_u)$, $P(s, a_d)$, $P(s, a_l)$, and $P(s, a_r)$ as shown in Fig.2(b). Then, we calculate a resultant vector $\overrightarrow{P}$ and an angle θ_o between $\overrightarrow{P}$ and a positional vector $\overrightarrow{L}_o$ for H_s. After that, an attention degree from self to the other $AD(H_o, H_s)$ is defined by:

$$AD(H_o, H_s) = (\cos\theta_o + 1)/2, \tag{10}$$

where $AD(H_o, H_s) \in [0, 1]$ holds.

4.2 Action Decision Using Attention Degree

To promote cooperative behavior, it is desired that the self agent approaches to the other agent having eye contact.

Using two attention degrees, i.e. $AD(H_s, H_o)$ in (9) and $AD(H_o, H_s)$ in (10), we choose a target agent t_a by

$$t_a = \arg\max_{i \in T_a} \frac{AD(H_i, H_s) \times AD(H_s, H_i)}{d(H_s, H_i)}, \tag{11}$$

where T_a refers to a set of subscript of other agents and $d(H_s, H_i)$ is a normalized distance between self H_s and other agent H_i ($i \in T_a$), i.e. $d(H_s, H_i) \in [0, 1]$.

In order to approach the self agent to the other agent having a higher attention value, Q-values should be recalculated by

$$Q'(s, a_k) = Q(s, a_k) \times (\cos\theta_k + 1)/2, \tag{12}$$

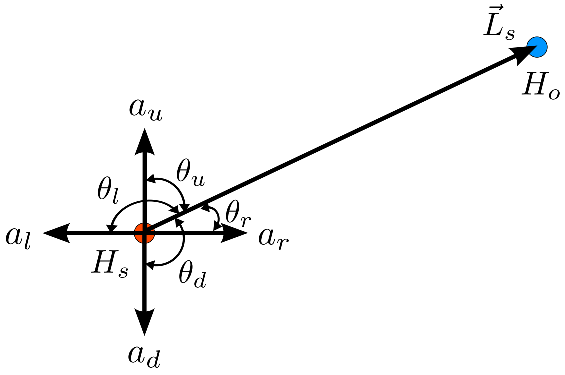


Fig. 3. Angeles between a positional vector $\overrightarrow{L}_s$ and directional vectors of action a_k

where a_k represents one of four possible actions in a grid world: a_u, a_d, a_l, or a_r and θ_k is an angle between a positional vector $\overrightarrow{L}_s$ and a directional vector of action a_k. θ_k is illustrated in Fig.3.

After that, agents select their actions based on the Boltzmann distribution (3).

Using the above action decision method, we expect to emerge cooperative behavior because agents try to cooperate with the other agent having eye contact.

5 Computer Simulation

In this section, through computer simulations using a pursuit problem, we verify the performance of the proposed intelligent learning system. At first, we describe a problem setting of the pursuit problem in Section 5.1. Secondly, we present a simulation setting in Section 5.2. Finally, we show simulation results in Section 5.3.

5.1 Problem Setting

A pursuit problem is a well-known multi-agent problem which plural hunters pursuit preys (or a prey) and catch them in a grid field. The followings are assumed in the present paper.

- Two dimensional 15×15 grid field with a torus structure.
- Six hunters H_i ($i \in \{1, 2, \cdots, 6\}$) and three preys P_j ($j \in \{1, 2, 3\}$) in the field. Initially, they are located randomly in the field.
- All the hunters can observe all the cells (complete observation) and act according to their own policy. The hunters can synchronously move up, down, left, or right by one cell, or stay on the same cell.
- All the preys can synchronously act according to a predefined policy.
- A goal state is assumed that each prey is occupied by any two hunters in two of four adjacent cells (up, down, left, and right). The two hunters can get a reward and the captured prey is removed from the field.

5.2 Simulation Setting

The hunters get a positive reward $r = 100$ if a goal state is reached. The number of steps is limited to 5,000 and we start a next trial if it reaches the limit.

The parameters were selected as learning rate $\alpha = 0.01$, discount rate $\gamma = 0.8$, the number of tiles $n = 1,500$, and positive parameter $\rho = 0.6$. The inverse temperature parameter was calculated by $\beta = 5.0 \times 10^{-4} e^{-t/100}$. Initial Q-values and P-values were set to 0.1 and 0.2, respectively. These parameters were selected so as to get the best performance through preliminary computer simulations.

5.3 Simulation Results

The learning curves is shown in Fig.4. In the figure, the horizontal axis represents the number of episodes and the vertical axis is the averaged number of steps. In the simulation, the number of steps is averaged for 5 trials. In this figure, red and green lines show learning curves for the proposed and the conventional systems, respectively. Here, the conventional system refers to a standard Q-learning system with random tile-coding. Note that a standard Q-learning system without random tile-coding could not tackle the given pursuit problem. As seen in Fig.4, the proposed method converges adequately but the conventional method does not converge at all. We conducted many computer simulations changing parameters and situations. It is verified that the proposed attention system works well in any cases. We observed that the proposed system promotes cooperative behavior.

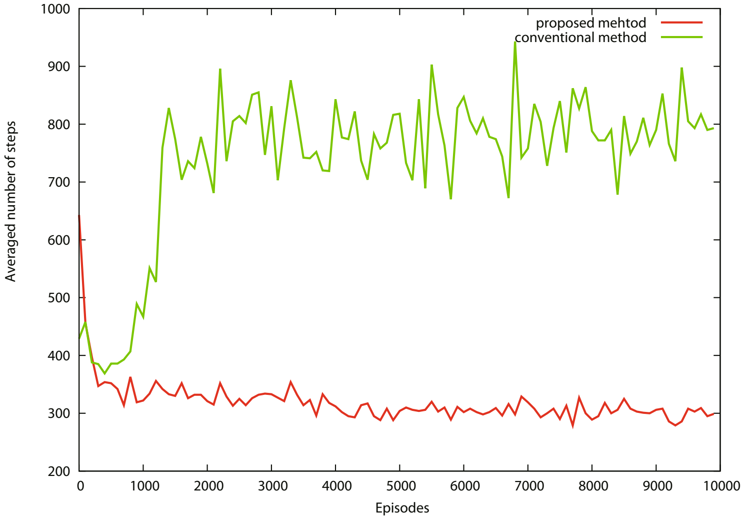


Fig. 4. Learning curves

6 Summary

In the present paper, we have proposed the proposed intelligent learning system using attention degree to emerge cooperative behavior in MASs. Firstly, we have introduced a concept of eye contact and formulated eye contact by a Q-value in reinforcement Q-learning method and a P-value in the policy estimation method. Secondly, we have proposed attention degrees both from self to the other and from the other to self. Thirdly, we have proposed an action decision method using the attention degrees that self agent approaches the other agent having eye contact. Finally, we have shown that the agents making eye contact each other pursue preys by approaching each other. Through computer simulations using a pursuit problem, we have verified that the proposed system has superior performance with the standard Q-learning system.

Acknowledgments. This work was partly supported by Grant-in-Aid for Scientific Research (No.23500181) from JSPS, Japan.

References

1. Stone, P., Veloso, M.: Multiagent Systems: A Survey from a Machine Learning Perspective. Autonomous Robots 8(3), 345–383 (2000)
2. Kaelbling, L.P., Littman, M.L., Moore, A.P.: Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research 4, 237–285 (1996)
3. Sutton, R. S., Barto, A. G.: Reinforcement Learning: An Introduction. MIT Press (1998)
4. Bratman, M.E.: Intention, Plans and Practical Reason. Harvard University Press (1987)
5. Nagayuki, Y., Ishii, S., Ito, M., Shimohara, K., Doya, K.: A Multi-Agent Reinforcement Learning Method with the Estimation of the Other Agent's Actions. In: Proceedings of the Fifth International Symposium on Artificial Life and Robotics, vol. 1, pp. 255–259 (2000)
6. Nagayuki, Y., Ito, M.: Reinforcement Learning Method with the Inference of the Other Agent's Policy for 2-Player Stochastic Games. Transactions on the Institute of Electronics, Information and Communication Engineers J86-D-I(11), 821–829 (2003) (in Japanese)
7. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning 8(3-4), 279–292 (1992)
8. Yokoyama, A., Omori, T., Ishikawa, S., Okada, H.: Modeling of Action Decision Process Based on Intention Estimation. In: Proceedings of Joint 4th International Conference on Soft Computing and Intelligent Systems and 9th International Symposium on Advanced Intelligent Systems, vol. TH-F3-1 (2008)
9. Yokoyama, A., Omori, T.: Model Based Analysis of Action Decision Process in Collaborative Task Based on Intention Estimation. Transactions on the Institute of Electronics, Information and Communication Engineers J92-A(11), 734–742 (2009) (in Japanese)
10. Kobayashi, K., Kanehira, R., Kuremoto, T., Obayashi, M.: An Action Selection Method Based on Estimation of Other's Intention in Time-Varying Multi-agent Environments. In: Lu, B.-L., Zhang, L., Kwok, J. (eds.) ICONIP 2011, Part III. LNCS, vol. 7064, pp. 76–85. Springer, Heidelberg (2011)
11. Hayashi, Y., Kojiri, T., Watanabe, T.: Focus Support Interface Based on Actions for Collaborative Learning. Neurocomputing 73(4-6), 669–675 (2010)

Basic Study on Particle Swarm Optimization with Hierarchical Structure for Constrained Optimization Problems

Kazuki Komori[1], Kazuhiro Homma[2], and Tadashi Tsubone[1]

[1] Nagaoka University of Technology, 1603-1, Kamitomiokamachi, Nagaoka,
Niigata, 940–2188, Japan
s081032@stn.nagaokaut.ac.jp, tsubone@vos.nagaokaut.ac.jp

[2] Information Technology Business Development Department, GaiaX Co. Ltd.,
KSS Gotanda Bldg., Nishi-Gotanda 1-21-8 , Shinagawa-ku, Tokyo, Japan

Abstract. In this work, we consider Particle Swarm Optimization (abbr. PSO) with hierarchical structures in order to solve some constrained optimization problems. The PSO with hierarchical structures has two layers. The lower layer is used to satisfy the constraint conditions and the upper layer is used to optimize the objective function. Due to these layers and the mutual function, the proposed method can be applied to constrained optimization problems which problems cannot be solved by the basic PSO. In this paper, we apply this procedure to some constrained optimization problems and evaluate its performance.

Keywords: Particle swarm optimization, constrained optimization problems, searching ability.

1 Introduction

In recent years, Particle Swarm Optimization (abbr. PSO) has attracted many researchers due to its ability to solve optimization problems. PSO is a swarm based stochastic optimization technique developed by Kennedy and Eberhart in 1995 with an idea taken from social behaviors of birds or fishes [1–3]. PSO optimizes many problems by some particles moving in the search-space. Each particle knows a position with the best value of the objective function which has found so far, and a position with the best value which tracked by whole swarm. These positions are called *pbest* and *gbest*, respectively. PSO searches an optimal solution of objective function based on the information of *pbest* and *gbest*.

PSO aims to optimize the single-objective function basically. Thereby, in order to solve constrained optimization problems, the objective function must be reformed to include constraint conditions such as a penalty method. The penalty method is well-known as an effective technique for solving constrained optimization problems. The penalty method transforms the objective function to an augmented objective function as including constraint terms and weight parameters.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 545–551, 2012.

In order to find an optimal solution which is satisfied the constraint conditions, weight parameters must be required tuning. However, the tuning is often pretty hard and needs heuristic procedure.

This paper proposes a novel PSO with hierarchical structures without use of augmented objective functions. This algorithm solves constrained optimization problems by using two layers. The lower layer ensures to satisfy constraint conditions. The upper layer operates to optimize the objective function. We apply this technique to some constrained optimization problems and evaluate its performance.

2 Constrained Optimization Problems

The optimization is to select the best from some alternatives. The optimization problem is reduced to find a decision variable x such that the objective function $f(x)$ makes the minimum or maximum. Many optimization problems entail constraint conditions in the real world. The constrained optimization problem is reduced to satisfy the constraint conditions and find a decision variable x that the objective function $f(x)$ makes the minimum.

$$\text{minimize} \quad f(\boldsymbol{x}) \quad , \ \boldsymbol{x} = [x_1, x_2, \ldots, x_n], \tag{1}$$

$$\text{subject to } g_k(\boldsymbol{x}) \leq 0, \ k = 1, \ldots, q, \tag{2}$$

$$h_k(\boldsymbol{x}) = 0, \ k = q+1, \ldots, m, \tag{3}$$

$$x_j^L \quad \leq x_j \leq x_j^U, \ j = 1, \ldots, n, \tag{4}$$

where n is the dimensionality of an objective function $f(\boldsymbol{x})$, $g_k(\boldsymbol{x})$ is inequality constraints, $h_k(\boldsymbol{x})$ is equality constraints, x_j^L is lower bound of x_j and x_j^U is upper bound of x_j. The feasible region of search satisfies all constraint conditions.

The penalty method can come down a constrained optimization problem to a basic optimization problem. The penalty method transforms the objective function to an augmented objective function including constraint terms and weight parameters.

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) + \lambda \left\{ \sum_{k=1}^{q} \max(0, g_k(\boldsymbol{x}))^2 + \sum_{k=q+1}^{m} h_k(\boldsymbol{x})^2 \right\}, \tag{5}$$

where λ is penalty factor. This penalty factor must be required tuning. However, the tuning is often pretty hard and needs heuristic procedure.

On next section, we propose a novel PSO without use of the augmented objective functions such as the penalty method.

3 Particle Swarm Optimization with Hierarchical Structures

In PSO, each particle knows a position with the best value of the objective function which has found so far, and a position with the best value of the objective function which tracked by whole swarm. These positions are called $\boldsymbol{pbest}_i$ and $\boldsymbol{gbest}$, respectively. PSO searches an optimal solution of objective function based on the information of $\boldsymbol{pbest}_i$ and $\boldsymbol{gbest}$. The particle's current position $\boldsymbol{x}_i$ can be considered as a set of coordinates describing a point in search-space of the objective function. The particle's position and velocity are updated by using the information of $\boldsymbol{pbest}_i$ and $\boldsymbol{gbest}$. The velocity and position at time t are updated by according to the following two equations respectively:

$$\boldsymbol{v}_i^{t+1} = \omega \boldsymbol{v}_i^t + c_1 \boldsymbol{rand}_1(\boldsymbol{pbest}_i^t - \boldsymbol{x}_i^t) + c_2 \boldsymbol{rand}_2(\boldsymbol{gbest}^t - \boldsymbol{x}_i^t), \tag{6}$$

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1}. \tag{7}$$

All particles share the $\boldsymbol{gbest}$ and are affected by the $\boldsymbol{gbest}$.

The basic PSO algorithm involves the following steps:

(Step1) The initial particle's positions $\boldsymbol{x}_i$ are assigned by uniform random numbers at $t = 0$. The particle's velocities $\boldsymbol{v}_i$ are initialized by zero.
(Step2) Each Particle's current position is evaluated by the objective function. If $f(\boldsymbol{x}_i) < f(\boldsymbol{pbest}_i)$, the current position has the best value of the objective function which has found so far that is, $\boldsymbol{pbest}_i$ is update to the particle's position $\boldsymbol{x}_i$.
(Step3) $\boldsymbol{gbest}$ is updated to $\boldsymbol{pbest}_i$ that is best position with the best value of the objective function in a set of $\boldsymbol{pbest}_i$.
(Setp4) The particle's position and velocity are updated according to equations (6) and (7).
(Step5) If iteration count t is less than T_{max}, return to (Step2).

The proposed method, PSO with hierarchical structures has two layers. The lower layer is used to satisfy the constraint conditions and the upper layer is used to optimize the objective function. Due to these layers and the mutual function, the proposed method can be applied to constrained optimization problems which problems cannot be solved by the basic PSO. Here, transforming inequality constraint into equality constraint to following.

$$g_k(\boldsymbol{x}) = |h_k(\boldsymbol{x})| - \epsilon \leq 0, k = q+1, \ldots, m, \tag{8}$$

where, ϵ is sufficiently small. A combined evaluation function which has all constraint conditions (1)-(4) is presented by following:

$$G(\boldsymbol{x}) = \sum_{k=1}^{m} \max(0, g_k(\boldsymbol{x})) + \sum_{j=1}^{n} \max(0, x_j^L - x_j) + \sum_{m=1}^{n} \max(0, x_m - x_m^U). \tag{9}$$

If $G(\boldsymbol{x}) > 0$, $\boldsymbol{x}$ isn't satisfying the constraint conditions and if $G(\boldsymbol{x}) = 0$, $\boldsymbol{x}$ is satisfying the constraint conditions. In order to satisfy the constraint conditions, PSO of the lower layer use the evaluation function (9).

Each particle knows a position with the best value of the evaluation function which found so far, and a position with best value which tracked by swarm of PSO of lower layer. These values are defined by $\boldsymbol{pCbest}_i$ and $\boldsymbol{sCbest}_l$, respectively. Also, each particle knows a position with the best value of the objective function under the condition of $G(\boldsymbol{x}) = 0$, and a position with best position which tracked by whole swarm of PSO of upper layer. These values are defined by $\boldsymbol{sObest}_l$ and $\boldsymbol{gObest}$, respectively. The particle's position and velocity are updated by according to the information of $\boldsymbol{pCbest}_i$, $\boldsymbol{sCbest}_l$, $\boldsymbol{sObest}_l$ and $\boldsymbol{gObest}$.

$$\begin{aligned} \boldsymbol{v}_i^{t+1} &= \omega \boldsymbol{v}_i^t + c_1 \boldsymbol{rand}_1(\boldsymbol{pCbest}_i^t - \boldsymbol{x}_i^t) + c_2 \boldsymbol{rand}_2(\boldsymbol{sCbest}_l^t - \boldsymbol{x}_i^t) \\ &\quad + c_3 \boldsymbol{rand}_3(\boldsymbol{sObest}_l^t - \boldsymbol{x}_i^t) + c_4 \boldsymbol{rand}_4(\boldsymbol{gObest}^t - \boldsymbol{x}_i^t), \qquad (10) \\ \boldsymbol{x}_i^{t+1} &= \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1}, \qquad (11) \end{aligned}$$

where $\boldsymbol{x}_i^t$ and $\boldsymbol{v}_i^t$ are position and velocity vector respectively. ω is an attenuation factor. c_1, c_2, c_3 and c_4 are the weight of each term. $\boldsymbol{rand}_1$, $\boldsymbol{rand}_2$, $\boldsymbol{rand}_3$ and $\boldsymbol{rand}_4$ are random numbers generated uniformly between [0,1].

The proposed algorithm involves the following steps:

(Step1) The initial particle's positions $\boldsymbol{x}_i$ are assigned by uniform random numbers at $t = 0$. The particle's velocities $\boldsymbol{v}_i$ are initialized by zero.
(Step2) To generate swarms. The number of swarms is C. All particles belong to the swarms. A set of these swarms makes PSO of the lower layer.
(Step3) Each particle's current position is evaluated by the evaluation function $G(\boldsymbol{x}_i)$. If $G(\boldsymbol{x}_i) \leq G(\boldsymbol{pCbest}_i)$, the current position has the best value of the evaluated function which has found so far that is, $\boldsymbol{pCbest}_i$ is updated to the particle's position $\boldsymbol{x}_i$.
(Step4) To compare $\boldsymbol{sCbest}_l$ and $\boldsymbol{pCbest}_i$ by the value of the evaluation function. If $G(\boldsymbol{pCbest}) \leq G(\boldsymbol{sCbest})$, a $\boldsymbol{sCbest}$ is updated to $\boldsymbol{pCbest}$. However, if there are two or more $\boldsymbol{pCbest}_i$ which satisfies $G(\boldsymbol{pCbest}_i) = 0$, $\boldsymbol{sCbest}_l$ is updated to randomly selected $\boldsymbol{pCbest}_i$ with the condition $G(\boldsymbol{pCbest}_i) = 0$.
(Step5) If $\boldsymbol{sCbest}_l$ which satisfies $G(\boldsymbol{sCbest}_l) = 0$ exists, $\boldsymbol{sCbest}_l$ is evaluated by the objective function. And to compare $\boldsymbol{sCbest}_l$ and $\boldsymbol{sObest}_l$ by the value of the objective function. If $f(\boldsymbol{sCbest}_l) < f(\boldsymbol{sObest}_l)$, $\boldsymbol{sObest}_l$ is updated to $\boldsymbol{sCbest}_l$. $\boldsymbol{gObest}$ is updated to $\boldsymbol{sObest}_l$ that is best position with the best value of the objective function in a set of $\boldsymbol{sObest}_l$.
(Step6) The particle's position and velocity are updated according to equations (10) and (11), where, if $G(\boldsymbol{sCbest}_l) \neq 0$, the c_3 and c_4 is to be zero.
(Step7) If all $\boldsymbol{sCbest}_l$ satisfy $G(\boldsymbol{sCbest}_l) = 0$, let each $\boldsymbol{sObest}_l$ compare by the value of the objective function. And remove M swarms which have worst evaluation value. And to regenerate M swarms by using same procedure of (Step1).
(Step8) If iteration count t is less than T_{max}, return to (Step3).

4 Experimental Results

The proposed particle swarm optimization with hierarchical structures is tested by using 13 benchmark problems on the literature [4]. Table 1 presents the characteristic of 13 benchmark problems. Type of f shows the characteristic of the objective function. ρ is percentage of feasible region of search on whole search-space. LI is the number of linear inequality constraints. NI is the number of nonlinear inequality constraints. LE is the number of linear equality constraints. NE is the number of nonlinear equality constraints.

Table 1. Characteristic of the 13 benchmark problems

Prob.	n	Type of f	ρ	LI	NI	LE	NE
g01	13	quadratic	0.0111%	9	0	0	0
g02	20	nonlinear	99.9971%	0	2	0	0
g03	10	polynomial	0.0000%	0	0	0	1
g04	5	quadratic	52.1230%	0	6	0	0
g05	4	cubic	0.0000%	2	0	0	3
g06	2	cubic	0.0066%	0	2	0	0
g07	10	quadratic	0.0003%	3	5	0	0
g08	2	nonlinear	0.8560%	0	2	0	0
g09	7	polynomial	0.5121%	0	4	0	0
g10	8	linear	0.0010%	3	3	0	0
g11	2	quadratic	0.000%	0	0	0	1
g12	3	quadratic	4.7713%	0	1	0	0
g13	5	nonlinear	0.0000%	0	0	0	3

g02, g03, g08 and g12 are maximization problems and others are minimization problems.

The parameters are set of $\omega = 0.729$ and $c_1 = c_2 = c_3 = c_4 = 0.9$. The number of particle i is equal to 300. The number of swarm of lower layer C is 30. The number of removing swarm M is 3. The allowable range of transformed inequality constraints from equality constraint, ϵ, is 0.0001. The maximum iteration count T_{max} is 1750. Table 2 presents experimental results of 30 trials. Optimal shows optimal solution. Table 3 presents experimental results using g Stochastic Ranking for Constrained Evolutionary Optimization h (abbr. SR) [4]. SR is based on penalty method.

SR can almost found optimal solution of benchmark problems. SR is superior to our proposed method on g03, g07, g09 and g13. However our proposed PSO has good performance similar to SR on otherwise problems. Note that SR doesn't have guaranteed to satisfy the constraint except problems finding the optimal solution, because SR is based on the penalty method. The solutions founded by our PSO with hierarchical structures are ensured to satisfy the constraint conditions for all problems.

Table 2. Experimental results on 13 benchmark functions using particle swarm optimization with a hierarchical structures

fcn	optimal	best	median	mean	st. dev.	worst
g01	-15.000	-15.000	-14.998	-14.654	7.9E-01	-12.436
g02	0.803619	0.785191	0.598371	0.591977	1.1E-01	0.386117
g03	1.000	0.945	0.871	0.859	5.3E-02	0.722
g04	-30665.539	-30665.539	-30665.539	-30665.539	4.8E-09	-30665.539
g05	5126.498	5126.501	5127.027	5128.178	3.5E+00	5145.376
g06	-6961.814	-6961.814	-6961.794	-6961.769	5.2E-02	-6961.599
g07	24.306	24.333	24.829	24.894	3.7E-01	26.003
g08	0.095825	0.095825	0.095825	0.095825	1.8E-17	0.095825
g09	680.630	680.632	680.672	680.676	3.6E-02	680.801
g10	7049.331	7054.112	7190.209	7247.643	2.0E+02	7909.672
g11	0.750	0.750	0.750	0.750	2.9E-06	0.750
g12	1.000000	1.000000	1.000000	1.000000	0.0E+00	1.000000
g13	0.053950	0.055302	0.199021	0.203331	1.1E-01	0.505264

Table 3. Experimental results on 13 benchmark functions using SR[4]

fcn	optimal	best	median	mean	st. dev.	worst
g01	-15.000	-15.000	-15.000	-15.000	0.0E+00	-15.000
g02	0.803619	0.803515	0.785800	0.781975	2.0E-02	0.726288
g03	1.000	1.000	1.000	1.000	1.9E-04	1.000
g04	-30665.539	-30665.539	-30665.539	-30665.539	2.0E-05	-30665.539
g05	5126.498	5126.497	5127.372	5128.881	3.5E+00	5142.472
g06	-6961.814	-6961.814	-6961.814	-6875.940	1.6E+02	-6350.262
g07	24.306	24.307	24.357	24.374	6.6E-02	24.642
g08	0.095825	0.095825	0.095825	0.095825	2.6E-17	0.095825
g09	680.630	680.630	680.641	680.656	3.4E-02	680.763
g10	7049.331	7054.316	7372.613	7559.192	5.3E+02	8835.655
g11	0.750	0.750	0.750	0.750	8.0E-05	0.750
g12	1.000000	1.000000	1.000000	1.000000	0.0E+00	1.000000
g13	0.053950	0.053957	0.057006	0.067543	3.1E-02	0.216915

5 Conclusions

In this paper, we proposed a novel procedure to solve some constrained optimization problems without the use of the augmented objective functions. The PSO method with hierarchical structures has two layers to satisfy the constraint conditions and to optimize the objective function respectively. The proposed method could be applied to constrained optimization problems. We confirmed the good performance of the proposed PSO by using some benchmark problems and by comparing to SR method based on penalty method.

In future, we will consider to extend our PSO in order to apply Multi-objective optimization problems.

References

1. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
2. Eberhart, R., Kennedy, J.: A New Optimizer Using Particle Swarm Theory. In: Sixth International Symposium on Micro Machine and Human Science, pp. 39–43. IEEE Service Center, Piscataway (1995)
3. Shi, Y.H., Russell, C.E.: Empirical Study of Particle Swarm OPtimization. Evolutionary Computation 3, 1945–1950 (1999)
4. Runarsson, T.P., Yao, X.: Stochastic Ranking for Constrained Evolutionary Optimization. IEEE Transactions on Evolutionary computation 4(3), 284–294 (2000)

A New Probabilistic Approach to Independent Component Analysis Suitable for On-Line Learning in Artificial Neural Networks

Marko V. Jankovic[1] and Neil Rubens[2]

[1] Electrical Engineering Institute "Nikola Tesla", Belgrade, Serbia
elmarkoni@ieent.org
[2] University of Electro-Communictaions, Tokyo, Japan
rubens@ai.is.uec.ac.jp

Abstract. Recently, elements of probabilistic model that are suitable for modeling of learning algorithms in biologically plausible artificial neural networks framework, have been introduced. Model was based on two of the main concepts in quantum physics – a density matrix and the Born rule. In this paper we will show that proposed probabilistic interpretation is suitable for modeling of on-line learning algorithms for Independent Component Analysis (ICA), which could be realized on parallel hardware based on very simple computational units. Proposed concept (model) can be used in the context of improving algorithm convergence speed, learning factor choice, input signal scale robustness, and can be easily deployed on parallel hardware.

Keywords: Probabilistic Independent Component Analysis, Born Rule, Tsallis Entropy, Local Learning Rules.

1 Introduction

Independent component analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements or signals. ICA defines model in which data variables are assumed to be linear or nonlinear mixtures of some unknown latent variables and the mixing system is also unknown. The latent variables are assumed mutually independent and nongaussian, and are referred to as the independent components of the observed data. These independent components can be found by ICA [1].

In this paper, we will propose a sort of "quantum" probabilistic model which relies on a very small number of assumptions and that is suitable, as we are going to demonstrate, for on-line learning algorithms. This probabilistic approach to PCA was recently proposed and analyzed in [2,4]. Here, we give definitions of probabilistic model for ICA calculation, which can be used to generate a number of different algorithms. The proposed concept (model) can be useful in the context of algorithm convergence speed, learning factor choice, or input signal scale robustness.

Why are we interested in a standard linear neural networks approach? Due to their low complexity, such algorithms and their implementation in neural networks are

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 552–559, 2012.

potentially useful for tracking of slow changes of correlations in the input data or for updating eigenvectors with new samples. Linear neural networks could also be implemented on highly parallel platforms like graphics processing units (GPU). Furthermore, these kind of networks can be used for calculations of general component analysis for the higher dimensional systems in comparison with solutions that require the storage of the whole covariance matrix. Finally, neural network approaches based on biologically plausible learning rules are still useful for research in which the goal is to make computational models that emulate some of the brain circuitry.

In this paper, we will define elements of probabilistic model that could be useful for general approach to on-line learning algorithms applied in neural networks context. From the aspect of artificial neural networks, the choice of different realization concepts has direct impact on the algorithm's convergence speed, preciseness, complexity of plausible hardware realization and biological plausibility. In Section 2, the Born rule is introduced. The Born rule in the artificial neural networks framework is introduced in Section 3. In section 4 we introduce new definitions for probabilistic ICA. Some small scale experimental results are presented in Section 5. Section 6 provides concluding remarks.

2 Quantum Probability Model and Born Rule

In this section, we give a short recapitulation of the quantum probability model and the Born rule, based on a similar section in [6, 2].

The quantum probability model takes place in a Hilbert space H of finite or infinite dimension. A state is represented by a positive semidefinite linear mapping (a matrix $\boldsymbol{\rho}$) from this space to itself, with a trace of 1, i.e. $\forall \boldsymbol{\Psi} \in$ H $\boldsymbol{\Psi}^T\boldsymbol{\rho}\boldsymbol{\Psi} \geq 0$, $\mathrm{Tr}(\boldsymbol{\rho}) = 1$. Such $\boldsymbol{\rho}$ is self adjoint and is called a density matrix.

Since $\boldsymbol{\rho}$ is self adjoint, its eigenvectors $\boldsymbol{\Phi}_i$ are orthonormal, and since it is positive semidefinite, its eigenvalues p_i are real and nonnegative $p_i \geq 0$. The trace of a matrix is equal to the sum of its eigenvalues, therefore $\sum_i p_i = 1$.

The equation $\boldsymbol{\rho} = \sum_i p_i \; \boldsymbol{\Phi}_i \quad \boldsymbol{\Phi}_i^T$ is interpreted as "the system is in state $\boldsymbol{\Phi}_i$ with probability p_i". The state $\boldsymbol{\rho}$ is called the pure state if $\exists i$ s.t. $p_i = 1$. In this case, $\boldsymbol{\rho} = \boldsymbol{\Psi}\boldsymbol{\Psi}^T$ for some normalized state vector $\boldsymbol{\Psi}$, and the system is said to be in state $\boldsymbol{\Psi}$. So, the most general density operator is in the form $\boldsymbol{\rho} = \sum_k p_k \; \boldsymbol{\Psi}_k \; \boldsymbol{\Psi}_k^T$ where the coefficients p_k are nonnegative and add up to one, and $\boldsymbol{\Psi}_k$ represent pure states. We can see that this decomposition is not unique.

A measurement M with an outcome z in some set $\boldsymbol{Z}$ is represented by a collection of positive definite matrices $\{\boldsymbol{m}_z\}_{z \in Z}$ such that $\sum_{z \in Z} \boldsymbol{m}_z = \mathbf{1}$($\mathbf{1}$ is being the identity matrix in H). Applying measurement M to state $\boldsymbol{\rho}$ produces the outcome z with probability

$$p_z(\boldsymbol{\rho}) = \mathrm{trace}(\boldsymbol{\rho}\boldsymbol{m}_z). \tag{1}$$

This is the Born rule. Most quantum models deal with a more restrictive type of measurement called the von Neumann measurement, which involves a set of projection operators $\boldsymbol{m}_a = \boldsymbol{a}\boldsymbol{a}^T$, for which $\boldsymbol{a}^T\boldsymbol{a}' = \delta_{aa'}$. In a modern language, von Neumann's measurement is a conditional expectation onto a maximal Abelian subalgebra of the algebra of all bounded operators acting on the given Hilbert space. As before, $\sum_{a\in M} \boldsymbol{a}\, \boldsymbol{a}^{\mathrm{T}} = \mathbf{1}$. For this type of measurement, the Born rule takes a simpler form: $p_{\boldsymbol{a}}(\boldsymbol{\rho})=\boldsymbol{a}^{\mathrm{T}}\boldsymbol{\rho}\boldsymbol{a}$. Assuming $\boldsymbol{\rho}$ is a pure state this can be simplified further to

$$p_{\boldsymbol{a}}(\boldsymbol{\rho}) = (\boldsymbol{a}^{\mathrm{T}}\boldsymbol{\Psi})^2. \tag{2}$$

So, we can see that, if the state is $\boldsymbol{\rho}$, the probability of the outcome of the measurement will be $\boldsymbol{a}$, which is actually defined by the cosine square of the angle between vectors $\boldsymbol{a}$ and $\boldsymbol{\Psi}$, or $p_{\boldsymbol{a}}(\boldsymbol{\rho})=\cos^2(\boldsymbol{a},\boldsymbol{\Psi})$.

3 Quantum Probability Model in Neural Networks Context

In this section, we recapitulate some quantum probabilistic concepts that can be used in a neural network framework. We show how neural networks can be used in a probabilistic framework that is basically based on the Born rule.

The basic single layer feedforward artificial neural network is depicted in Fig. 2. The output of the n-th output unit y_n (n=1, 2,…, N) of a layer of parallel linear artificial neurons is given as

$$y_n(i) = \boldsymbol{w}_n(i)^{\mathrm{T}}\boldsymbol{x}(i),$$

with $\boldsymbol{x}(i)$ denoting a K-dimensional zero-mean input vector of the network and $\boldsymbol{w}_n(i)$ denoting a weight vector of the n-th output unit, and i represents sampling instances iT, where T is a sampling period. The output vector $\boldsymbol{y}$ is defined as

$$\boldsymbol{y}(i) = \boldsymbol{W}(i)^{\mathrm{T}}\boldsymbol{x}(i). \tag{3}$$

In the usual interpretation, based on specific requirements, e.g. minimization of some cost function, matrix $\boldsymbol{W}$ is changed (trained) in the process of learning, according to some adopted learning rule.

Here we will give a slightly different interpretation. We will consider a Hilbert space H of a finite dimension. “State vectors” are defined by the input data vector $\boldsymbol{x}_k$ and we can imagine that every vector $\boldsymbol{x}_k$ is available in a big enough number of copies (clones), so that we can perform as many simultaneous measurements as we want. A measurement M with an outcome $\boldsymbol{w}_n$ in some set $\boldsymbol{W}$ is represented by a collection of positive definite matrices $\{\boldsymbol{m}_{wn}\}_{wn\in W}$ such that $\boldsymbol{m}_{wn}=\boldsymbol{w}_n\boldsymbol{w}_n^{\mathrm{T}}$, so $\sum_{wn\in W} = \boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}$, which is not necessarily equal to the identity matrix on H. This means that the sum of the probabilities of the particular outcomes does not have to be equal to one – in other words, sometimes we will work with improper discrete distributions. Also, measures like entropy and divergence will be applied to improper probability distributions, or to a mixture of proper and improper probability distributions. In the following sections, we will point out that in the adopted framework, this will not affect the final result.

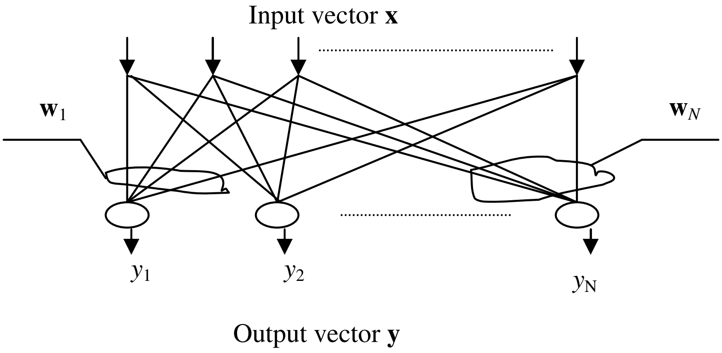


Fig. 1. A layer of parallel linear artificial neurons

Applying measurement M to state $\boldsymbol{x}_k$ produces outcome $\boldsymbol{w}_n$ with the probability (the Born rule)

$$p(\boldsymbol{w}_n \mid \boldsymbol{x}_k) \overset{def}{=} \cos(\boldsymbol{w}_n, \boldsymbol{x}_k),$$

regardless of the norm of the vectors $\boldsymbol{w}_n$ and $\boldsymbol{x}_k$. In the following text, we will consider only vectors $\boldsymbol{w}_n$ that have unit norms. This means

$$p(\boldsymbol{w}_n, \boldsymbol{x}_k) = \frac{\left(\boldsymbol{w}_n^{\mathrm{T}} \boldsymbol{x}_k\right)^2}{\left\|\boldsymbol{x}_k\right\|^2}. \tag{4}$$

Also, if we apply N simultaneous measurements M^N to the state $\boldsymbol{x}_k$ we obtain outcome $\boldsymbol{W}$ with the probability

$$p(\boldsymbol{W} \mid \boldsymbol{x}_k) \overset{def}{=} \sum_{n=1}^{N} p(\boldsymbol{w}_n \mid \boldsymbol{x}_k). \tag{5}$$

Here, it is assumed that the outcome of each measurement is different. We define the joint probability of the state $\boldsymbol{x}_k$ and outcome $\boldsymbol{W}$ obtained by simultaneous multiple measurement M^{N} on state $\boldsymbol{x}_k$, $p(\boldsymbol{W}, \boldsymbol{x}_k)$ as

$$p(\boldsymbol{W}, \boldsymbol{x}_k) \overset{def}{=} p(\boldsymbol{W} \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k). \tag{6}$$

Now, without loss of generality, let's assume that we are dealing with a random variable $\boldsymbol{x}$ that takes realizations from a set of observed K-dimensional zero-mean data vectors $\{\boldsymbol{x}_k\}$, $k \in \{1, \ldots, N_{\text{sample}}\}$, which are sampled from some distribution in time instants $t = kT$ where k is already defined and T represents the sampling period. Then, we can define $p(\boldsymbol{x}=\boldsymbol{x}_k \mid t=kT)$ as

$$p(\boldsymbol{x}_k) \overset{def}{=} \frac{\|\boldsymbol{x}_k\|^2}{\sum_{i=1}^{N_{sample}} \|\boldsymbol{x}_i\|^2}, \tag{7}$$

where N_{sample} represents the overall number of samples that are going to be analyzed. It is interesting to note that the only thing that we can conclude about the $p(\boldsymbol{x}_k)$ is that it is proportional to $\|\boldsymbol{x}_k\|^2$. The sum in the denominator represents the energy of samples that are going to be analyzed – we actually do not know the value of that sum at any, but the final moment. However, we know that it represents some constant. We can easily see that the adopted probability measure fulfils the two conditions that are required for the probability function $f(z)$ (in our case $p(z)$) to be considered as a modified generalized probability measure [5]:

1. For each z, $0 \leq f(z) \leq 1$,
2. $\sum_i f(z_i) = 1$.

In this definition, orthonormallity is not explicitly required in order that the coefficients $f(z_i)$ sum up to one. However, from the JSPS introduction [4], we can see that it is always implicitly present.

Here, we will consider all vectors as "oriented energies" or

$$\boldsymbol{x}_k = \|\boldsymbol{x}_k\| \frac{\boldsymbol{x}_k}{\|\boldsymbol{x}_k\|} = \|\boldsymbol{x}_k\| \boldsymbol{x}'_k ,$$

where the norm of the vector $\|\boldsymbol{x}_k\|$, represents the square root of the energy contained in the vector $\boldsymbol{x}_k$, and the orientation represents some unit norm vector $\boldsymbol{x}'_k$, which represents some pure state. In that case, we can see that the statistical description of our system is represented by the density matrix $\boldsymbol{\rho}$

$$\boldsymbol{\rho} = \sum_k p_k \boldsymbol{x}'_k \boldsymbol{x}'^{\mathrm{T}}_k ,$$

as a statistical mixture of pure states $\boldsymbol{x}'_k$, and $p_k = p(\boldsymbol{x}_k)$ are defined by (7). We have to stress that the density matrix $\boldsymbol{\rho}$ that is created here, does not fulfill the requirements of quantum mechanical postulates, since it connects the pure states from different time instants. However, we used this term here to stress the conceptual analogy with original definition of density matrix (although we could create a new term – e.g. normalized covariance matrix). We can see that

$$\boldsymbol{\rho} = \frac{N_{sample}}{\sum_{i=1}^{N_{sample}} \|\boldsymbol{x}_i\|^2} \boldsymbol{C},$$

where $\boldsymbol{C}$ is input signal covariance matrix. Obviously, the matrix $\boldsymbol{\rho}$ and the matrix $\boldsymbol{C}$ have the same eigenvectors.

In the proposed context, the learning algorithm applied to the neural network has a basic task - to find the measurement system in which input data is "best explained", or have the features that are specified. As an example, principal component analysis will search for the measurement (or we can say coordinate) system in which the input data covariance matrix is diagonal.

4 Probabilistic Independent Component Analysis

In this section, we are going to give a definition of the probabilistic ICA that can be used for creation of symmetric ICA algorithms. There is also another possibility to create asymmetric algorithms as done in [3].

Definition 1: ICA can be defined as a problem of minimization and (or) maximization of the entropy (like Tsallis or Shannon entropy) of the joint probability distribution $p(\boldsymbol{W}, \boldsymbol{x})$ of the input signal $\boldsymbol{x}$ and outcome $\boldsymbol{W}$, obtained by the simultaneous multiple measurement M^N on state (input signal) $\boldsymbol{x}$, under the constraint that the matrix $\boldsymbol{W}$ is orthonormal and $\boldsymbol{x}$ represents prewhitened signal. So, we have to solve following constrained problem (in the case of Tsallis entropy of q-th degree)

$$\underset{\mathbf{W}}{\min\, and/or \max} \mathrm{E}\left(\frac{1-\sum_{k=1}^{K} p(\mathbf{W}, x_k)^q}{q-1} \right), \tag{8}$$

under constraint that $\boldsymbol{W}$ is orthonormal matrix. We can see that the proposed probabilistic definition of ICA, requires maximization and (or) minimization of the entropy. This means that final algorithm depends on the type of the signals that are going to be retrieved. Also, we can notice that quadratic nonlinearity cannot be used for signal separation. In the case (when $q \to 0$) Tsallis entropy will become Shannon entropy and that function can be used for signal separation.

5 Simulation Results

Now we will examine the small scale numerical simulations results. The number of inputs was $K = 4$ and the number of output neurons was $N = 4$. Artificial zero-mean vectors with uncorrelated elements were generated by the following equations:

$$s(1,i) = .45\sin(2\pi \cdot i/5);$$
$$s(2,i) = .15\,((\mathrm{rem}(i,23)-11)/9).\hat{}5;$$
$$s(3,i) = .45\sin(2\pi \cdot i/37);$$
$$s(4,i) = .15\,((\mathrm{rem}(i,31)-15)/13).\hat{}5;$$

Input signal is constructed as $z = \boldsymbol{mix} * \boldsymbol{s}$, where mixing matrix mix is defined as

$$\mathbf{mix} = -.5 + \operatorname{rand}(K),$$

and after prewhitening signals were introduced as inputs ($\boldsymbol{x}$) to neural network. In Fig.2 we can see results of extraction of independent components after we minimized Tsallis entropy for q=0.5. We can see expected results, that some signals (in this case supergaussians) are satisfactorily extracted, while we were not able to separate subgaussian signals.

Fig. 3 represents results of independent components extraction after we minimized Tsallis entropy for q=2. Again, we had successful extraction of supergaussians and we were not able to separate subgaussian signals.

If we performed maximaization of Tsallis entropy for q=2 and q=0.5 we would, in both cases, successfully extract subgaussian signals and not be able to separate supergaussians.

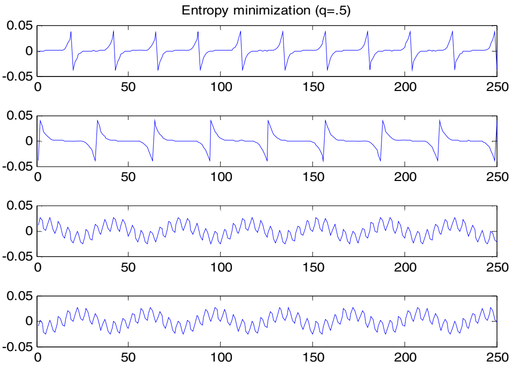


Fig. 2. Blind signal extraction of deterministic components (Tsallis entropy minimization, q=.5)

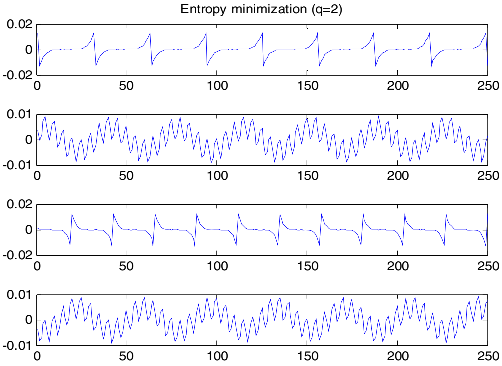


Fig. 3. Blind signal extraction of deterministic components (Tsallis entropy minimization, q=2)

By selecting any other q, or by selecting different entropy function we can have signal separation, but speed convergence and preciseness will be different, as well as possibility for successful implementation on parallel hardware, which is of great interest in high dimensional cases.

6 Conclusion

In this paper, we proposed a new, “quantum”, probabilistic ICA model that could be useful for implementation in on-line learning neural networks context. Model is based on the Born rule. Here we only considered derivation of symmetric algorithms. By selecting different entropy functions, it is possible to create a large number of ICA algorithms. This makes it possible to create algorithms that could be optimal from the point of view of convergence speed, preciseness, complexity of hardware implementation, locality of calculations, etc. With proposed probabilistic definition, ICA can be used in semi-supervised context and can be successfully implemented in parallel computation machines.

Acknowledgements. This work was supported in part by Serbian Ministry of Science and Education under grant TR33020.

References

1. Hyvarinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley and Sons (2001)
2. Jankovic, M., Rubens, N.: A new probabilistic approach to on-line learning in artificial neural networks. In: ASMCSS 2009, Greece, pp. 226–231 (2009)
3. Jankovic, M., Sugiyama, M.: A Multipurpose Linear Component Analysis Method Based on Modulated Hebb-Oja Learning Rule. IEEE Signal Processing Letters 15, 677–680 (2009)
4. Jankovic, M., Sugiyama, M.: Probabilistic principal component analysis based on JoyStick Probability Selector. In: IJCNN 2009, Atlanta, USA, pp. 1414–1421 (2009)
5. Warmuth, M.K., Kuzmin, D.: Bayesian generalized probability calculus for density matrices. Machine Learning 78, 63–101 (2010)
6. Wolf, L.: Learning using the Born rule. Technical report MIT-CSAIL-TR-2006-036 (2006)

Immune Algorithm for Bitmap Join Indexes

Amina Gacem[1] and Kamel Boukhalfa[2]

[1] National High School of Computer Science, Algiers
[2] Algeria Houari Boumediene University of Science and Technology, Algiers, Algeria
a_gacem@esi.dz, kboukhalfa@usthb.dz

Abstract. Bitmap join indexes are designed to prejoin the facts and dimension tables in data warehouses modeled by a star schema. They are defined on the fact table using attributes which belong to one or many dimension tables. The index selection process has become an important issue regarding the complexity of the search space to explore. Thus, the indexes can be defined on several attributes from several dimension tables (that may contain hundreds of attributes). However, only a few selection algorithms were proposed. In this article, we present a bitmap join indexes selection approach based on artificial immune algorithm. An experimental study was conducted on the dataset generated from APB-1 benchmark in order to compare the artificial immune algorithm with other algorithms.

Keywords: Artificial Immune System, Bitmap Join Indexes, Data Warehouses.

1 Introduction

The data warehouses (DW) are generally modeled by a star schema which contains a central and large fact table, and dimension tables describing the facts [1]. Data warehouses are used in an online analysis processing (OLAP) to perform complex decision queries. These queries require the execution of a multitude of joins between the fact and dimension tables, therefore making the execution more costly. The cost becomes more prohibitive when huge data are accessed by those queries. So, in order to reduce the execution time of queries, the Data Warehouse Administrator (DWA) has to optimize them. The query optimization is obtained by selecting optimization structures during the physical design phase. Indexes have already shown their performance in the traditional databases. We can mention B-trees and their variants [2], and join indexes [3]. However, the indexing techniques used in databases are not adapted to data warehouse environments [4]. Therefore, many indexation techniques dedicated to data warehouses have emerged like bitmap indexes [5], star join indexes [6] and the bitmap join indexes (BJI) [7] which best fit the DW because they optimize the star joins and selection operations defined on dimension tables. When selecting a BJI, many configurations are possible. We can create as much indexes as existing attributes in the dimension tables. However, indexes require enough space storage, so not

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 560–567, 2012.

all attributes can be indexed. Thus, only indexes that improve significantly the queries performance must be chosen. We propose in this article to use complex algorithms, more specifically, artificial immune algorithms, which have been applied in a wide range of computing fields. This paper is organized as follows: section 2 explains the BJI selection problem and its principle. Then, in section 3, we manage to offer glimpses of how the AIS' algorithms work and their applications. Section 4 presents our BJI selection approach with specific details for each complex algorithm. Following that, in section 5, we describe our experimental study to compare the results obtained by the algorithms. We conclude the paper in the section 6.

2 Bitmap Join Indexes

BJI are defined on the fact table using attributes that reference one or many dimension tables in order to make the join operations more efficient in the star schema. A bitmap representing the fact table's rows is created for each distinct value of the attribute that belongs to the dimension table on which the index is defined. The bit I of the bitmap equals 1 if the row that corresponds to the value of the indexed attribute can be joined with the row I of the fact table. Otherwise, the bit equals 0.

The binary nature of BJI improves query performance by allowing to apply logical operations AND, OR, NOT, etc. BJI are also very helpful for *count(*)* queries since only BJI have to be interrogated to answer those queries. We can illustrate this property in the example below: figure 1 represents a star schema with the fact table sales and the dimension table customer. Let's have the query Q1:

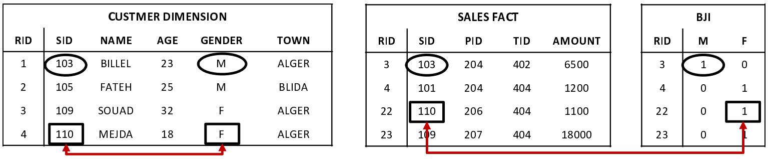
```
SELECT count(*)FROM Sales S,Customer C WHERE S.SID=C.SID AND C.GENDER='F'
```

To improve execution time of this query, the DWA creates a BJI on the attribute gender with the following SQL command:

```
CREATE BITMAP INDEX BJI_Gender
ON Sales (Customer.Gender) FROM Sales S, Customer C WHERE S.SID=C.SID
```

When executing the query Q1, the optimizer reads the bit vectors associated to the value 'F' and computes the number of '1' in the result vector.

The BJI Selection Problem (BJISP) is known NP-complete [8][9]. We can formalize the problem as follows: given a DW with d dimension tables $D = \{D_1, D_2, ..., D_d\}$ and a fact table F, a workload $Q = \{Q_1, Q_2, ..., Q_m\}$ where each query has a frequency Fj , a set of indexable candidates attributes $AS = \{A_1, A_2, ..., A_n\}$ and a storage space S. The BJISP intends to select a BJI configuration CI that reduces the execution cost of Q and does not exceed the storage bound. If the DWA wishes to select one index amongst n indexable attributes, he must evaluate $2^n - 1$ possible configurations [10]. To select several BJI defined on one or more attributes, he must evaluate $2^{2^{n-1}} - 1$ possible configurations [10].



CUSTMER DIMENSION					
RID	SID	NAME	AGE	GENDER	TOWN
1	103	BILLEL	23	M	ALGER
2	105	FATEH	25	M	BLIDA
3	109	SOUAD	32	F	ALGER
4	110	MEJDA	18	F	ALGER

SALES FACT				
RID	SID	PID	TID	AMOUNT
3	103	204	402	6500
4	101	204	404	1200
22	110	206	404	1100
23	109	207	404	18000

BJI		
RID	M	F
3	1	0
4	0	1
22	0	1
23	0	1

Fig. 1. An example of bitmap join index

Many BJI are eligible, so the Data Warehouse Administrator (DWA) has to choose one configuration, which is a complex task.There is a multitude of work that aim to automate the selection of BJI, they consist of two phases: (1)*pruning search space* to reduce the selection problem's complexity by using data mining algorithms [8][11], or algorithms based on other optimization structures such as horizontal partitioning [4][12] and (2)*Execution of algorithms* to determine a final configuration of indexes [8][10][4][13]. The algorithms used to determine a final configuration of indexes can be divided into two categories: *greedy algorithms* and *complex algorithms* such as heuristics, genetic algorithms, and data mining algorithms. The cost-based greedy algorithms used to select a configuration of BJI were the subject of many works [8][10][4]. The selection of a BJI using a genetic algorithm was mentioned in [13]. Our analysis of the literature leads us to conclude only a few works focus on complex algorithms to determine a BJI configuration.

3 Artificial Immune System (AIS)

The Artificial Immune System (AIS) is a meta-heuristic that combines features of natural immune systems such as memorization, learning and adaptations. The immunity is also a learning process through its ability to recognize threats, to memorize past attackers and its faculty of adaptation. The AIS had been first introduced in 1986 by [14]. The authors aim to observe the immune system's behavior by a simulation on a computer in order to gain a better understanding of immunity in real organisms. However, they notice major similarities between the AIS and the classifier system which leads them to believe that the immune system model can be used to perform artificial intelligence tasks, in particular learning tasks. Afterwards, many authors contributed significantly to enhance the knowledge about the AIS.

Unlike some other bio-inspired techniques, such as genetic algorithms and neural networks, the field of AIS encompasses a spectrum of algorithms because different algorithms implement different properties of different cells. All AIS algorithms mimic the behavior and properties of immunological cells, specifically B-cells, T-cells and dendritic cells (DCs), and the resultant algorithms exhibit differing levels of complexity and can perform a range of tasks [15]. The oft-cited AIS models are : *Negative Selection Algorithms* [16], *Clonal Selection Algorithm* [17], *Immune Network Models* [18]. The AIS was used to resolve many problems

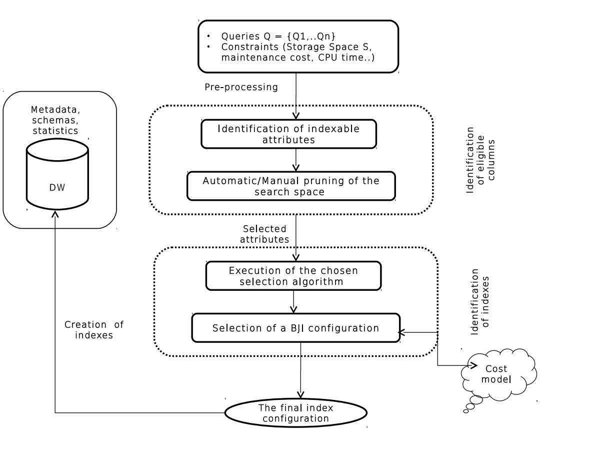

• Queries Q = {Q1,..Qn}
• Constraints (Storage Space S, maintenance cost, CPU time..)
Pre-processing
Metadata, schemas, statistics
DW
Identification of indexable attributes
Automatic/Manual pruning of the search space
Identification of eligible columns
Selected attributes
Execution of the chosen selection algorithm
Selection of a BJI configuration
Identification of indexes
Creation of indexes
Cost model
The final index configuration

Fig. 2. Our Selection Approach Architecture

related to a wide variety of fields: *Computer Security* [16], *Anomaly Detection* [19], *Fault Diagnosis* [20], *Data Mining and Retrieval* [21], *Adaptive Control* [22], *Web Mining* [23].

As we have seen it above, many complex algorithms and meta-heuristics have been used to perform an efficient physical design of data warehouses. Nevertheless, no author has ever applied the AIS in that field. So we believe that the use of AIS to perform physical design tasks relating to BJI has been the subject of academic work for the first time in this article.

4 A New Approach of Selection Based on Artificial Immune System

The immune learning algorithm requires the use of antigens as learning data, the system has to produce antibodies. In the context of our work, we have considered BJI as antibodies, and the queries as antigens. The general schema of the selection by AIS is illustrated in the figure 2. The selection consists of two phases: initialization and antigen presentation. In the first phase, an initial BJI configuration is generated randomly. In the second phase, a succession of immune operators is applied iteratively to sharpen the configuration.

Initialization. Each BJI is coded as a series of numbers where each number represents an attribute with a size that cannot exceed the number of indexable attributes. Suppose the set of indexable attributes consists of 9 attributes: *A0 : Retailer, A1 : Line, A2 : Year, A3 :Quarter, A4 : Week, A5 : Class, A6 : Division, A7 : Day, A8 : All, A9 :Group*. For example, we have an BJI coded as (4,9,1), which means the index is defined on *Week, Group* and *Line*. Let POP be the list of BJI to create.

Antigen Presentation. For each query Qi, do:

- *Clonal selection and expansion* : computes the affinity of each BJI of the POP list with the query R by building a query-attribute matrix MUA. If the affinity overtakes a threshold defined at the beginning, the BJI will be elected and duplicated following the affinity value. Thus, BJI that appears most in the maximum of queries will be the most duplicated index. Let CLN be the list of chosen and cloned BJI.
- *Maturation affinity* : each clone in CLN is muted inversely to its affinity. The number of mutations to do equals the size of BJI - its affinity. In the previous example, the size of BJI (4,9,1) equals 3 and the affinity is 1. Hence, the number of mutation is 3 - 1= 2.
- *Clonal interactions* : represents in our problem the network interaction or affinities between BJI. The affinity between two BJI is computed by summing the affinities between all their attributes, after building an attributes affinity matrix AAM.
- *Clonal deletion* : removes BJI that have all their affinities with other BJI inferior to predefined threshold (computed experimentally) and store the rest of indexes in a MEM list. To define the affinity between a BJI and other BJI, we sum the affinities of the BJI with others indexes that belong to MEM.
- *Meta-dynamic* : eliminates the BJI that have their affinity with antigen Qi inferior to predefined threshold from MEM (the affinity of BJIi with Qi is already defined in the point a).
- *Network Construction*: incorporates the remaining BJI of MEM with the BJI of the network, this new list is RES, and it was initially empty.
- *Network Interactions* : determines the similarity between each pair of BJI of the network from the matrix AAM by computing the affinity between two BJI as seen in point c.

Cycle. These steps until the end.

MEM contains the BJI selected according to a query Qi. RES contains the best elements of MEM, and then, every MEM query will be initialized at null value, in contrast with RES which will be initialized at the launch of the process.

5 Experiments

We conduct an experimental study to compare our algorithm based on AIS with other well-known approaches that have already been tested in previous works: genetic algorithms and K-mean by using the mathematical cost model defined in [8]. Hence, we have tested these algorithms on an Intel machine Core I3 with 3 GB of memory and storage disks of 500 GB. On this machine we have installed an APB1 [24] benchmark with Oracle DBMS 11g. The APB benchmark contains a star schema with a fact table *Actvars (24 786 000 rows), Prodlevel (9000 rows), Custlevel (900 rows), Timelevel (24 rows) and Chanlevel (9 rows).* We decide to run the five most frequent queries.

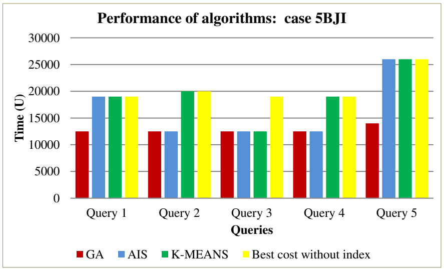


Fig. 3. Performance of proposed algorithms : case 5 *BJI*

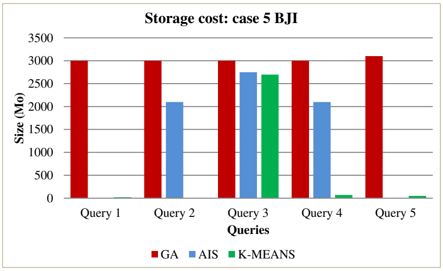


Fig. 4. Storage cost : case 5 *BJI*

In our experiment, we execute each selection algorithm: GA, AIS, K-means under a constraint on the storage space = 3 GB (parameters of GA are: crossover rate = 1, Mutation Rate = 0.3, size of population = 50, number of generations = 50). Each algorithm generates a set of 5 BJI. In the first test, for each query and each selection algorithm, we measure the execution cost of each query using BJI selected by each algorithm (figure 3) and the storage cost of generated BJI costs (figure 4). From the figure 3, we observe that the genetic algorithm generates a configuration that makes it possible for every query to run faster. But the storage rate (figure 4) of indexes created by GA is dramatically higher than storage rates of indexes obtained with other algorithms (AIS and K-means).

So the GA generates large BJI which reduce the execution cost of queries. Consequently, a compromise has to be made between both execution and storage cost. To achieve that, we have defined a variable that combines these two parameters by multiplying the execution and storage costs. The results are presented in figure 5. K-means and AIS algorithms give better results than GA. To test the efficiency of the algorithms when the constraint on the storage space is relaxed, we set a value of 5 GB to the storage space. The execution of every algorithm has induced to the creation of 10 BJI. The figures 6 and 7 show respectively the execution cost of queries in the presence of 10 BJI produced by each algorithm and the ratio of execution cost to the storage cost of BJI. We notice

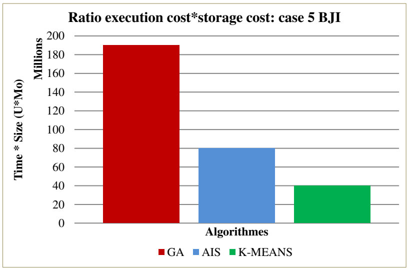


Fig. 5. Ratio execution cost*storage cost: case 5BJI

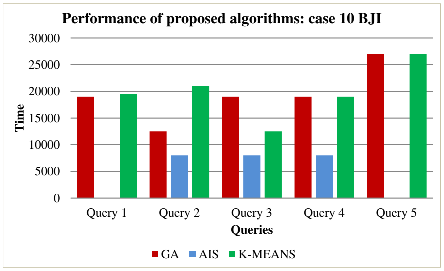


Fig. 6. Performance of proposed algorithms: case 10 BJI

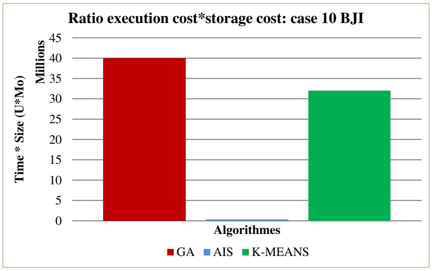


Fig. 7. Ratio execution cost*storage cost: case 10BJI

that the immune algorithm AIS has generated a configuration that reduces the execution cost of all queries while optimizing the required storage space. This algorithm provides better results than GA and K-means. Thus, the DWA can use it in the case he has enough storage space available to implement indexes.

6 Conclusion

In this paper, we focus on the optimization of complex queries defined on star schema DWs and propose a new approach based on immune algorithms to select BJI. Then, we have compared our approach with other classical algorithms: genetic algorithms and datamining algorithm (K-means). All the algorithms aim to reduce the time needed to execute the queries load without any violation of the condition on storage space by using mathematical cost model. The results clearly show that the AIS offers the better ratio execution cost* storage cost. We suggest as a continuation of this work (1) configure empirically the parameters of the algorithms to achieve better results, (2) introduce an intelligent agent that prunes automatically the attributes and the queries, (3) integrate these approaches with others optimization techniques such as horizontal and vertical partitioning and (4)consider a large workload of queries to scaling.

References

1. Kimball, R., Strehlo, K.: Why decision support fails and how to fix it. SIGMOD Record 24(3), 92–97 (1995)
2. Comer, D.: The ubiquitous b-tree. ACM Comput. Surv. 11(2), 121–137 (1979)
3. Valduriez, P.: Join indices. ACM Transactions on Database Systems 12(2), 218–246 (1987)
4. Bellatreche, L., Boukhalfa, K., Mohania, M.: Pruning Search Space of Physical Database Design. In: Wagner, R., Revell, N., Pernul, G. (eds.) DEXA 2007. LNCS, vol. 4653, pp. 479–488. Springer, Heidelberg (2007)
5. Chan, C.Y., Ioannidis, Y.E.: Bitmap index design and evaluation. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 355–366 (June 1998)

6. Red Brick Systems Inc., Star schema processing for complex queries. White Paper (1997)
7. O'Neil, P., Graefe, G.: Multi-table joins through bitmapped join indices. SIGMOD Record 24(3), 8–11 (1995)
8. Aouiche, K., Boussaid, O., Bentayeb, F.: Automatic Selection of Bitmap Join Indexes in Data Warehouses, pp. 64–73 (2005)
9. Boukhalfa, K., Bellatreche, L., Ziani, B.: Index de jointure binaires: Stratgies de slection et etude de performances. In: Journe Francophone sur les Entrepts de donnes et l'Analyse en ligne (EDA 2010). Revue des Nouvelles Technologies (2010)
10. Bellatreche, L., Boukhalfa, K.: Yet Another Algorithms for Selecting Bitmap Join Indexes. In: Bach Pedersen, T., Mohania, M.K., Tjoa, A.M. (eds.) DAWAK 2010. LNCS, vol. 6263, pp. 105–116. Springer, Heidelberg (2010)
11. Bellatreche, L., Missaoui, R., Necir, H., Drias, H.: A data mining approach for selecting bitmap join indices. Journal of Computing Science and Engineering 2(1), 206–223 (2008)
12. Stöhr, T., Martens, H., Rahm, E.: Multidimensional database allocation for parallel data warehouses. In: International Conference on Very Large Databases, pp. 273–284 (2000)
13. Bouchakri, R., Bellatreche, L., Boukhalfa, K.: Slection statique et incrmentale des index de jointure binaires multiples. In: Journe Francophones sur les Entrepts de donnes et l'Analyse en ligne (EDA 2011). Revue des Nouvelles Technologies RNTI, France (2011)
14. Farmer, J., Packard, N., Perelson, A.: The immune system, adaptation and machine learning. Physica 22D, 187–204 (1986)
15. Ishida, Y.: Immunity-Based-Systems: A Design Perspective. Springer (2004)
16. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: IEEE Symposium on Research in Security and Privacy (1994)
17. De Castro, L., Von Zuben, F.: Learning and optimization using the clonal selection principle. IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Intelligence Systems, 239–251 (2002)
18. Timmis, J., Neal, M., Hunt, J.: An artificial immune system for data analysis. Knowledge Based Systems 14(3-4), 121–130 (2000)
19. Dasgupta, D., Forrest, S.: Novelty detection in time series data using ideas from immunology. In: 5th International Conference on Intelligent Systems (1996)
20. Ishida, Y.: An Immune Network Model and its Applications to Process Diagnosis Systems and Computers (1993)
21. Hunt, J., Cooke, D., Holstein, H.: Case memory and retrieval based on the immune system. In: International Conference on Case Based Reasoning (1995)
22. Bersini, H.: Immune network and adaptative control. In: First European Conference on Artificial Life (1991)
23. Nasaroui, O., Dasgupta, D., Gonzlez, F.: A novel artificial immune system approach to robust datamining. In: Genetic and Evolutionary Computation Conference (2002)
24. Council, O.: Apb-1 olap benchmark, release ii (1998), http://www.olapcouncil.org/research/bmarkly.htm

Data Driven System Identification Using Evolutionary Algorithms

Awhan Patnaik, Samrat Dutta, and Laxmidhar Behera

Indian Institute of Technology, Kanpur, India
{awhanp,samratd,lbehera}@iitk.ac.in

Abstract. We present an evolutionary algorithm(EA) based system identification technique from measurement data. The nonlinear optimization task of estimating the premise parameters of a Takagi-Sugeno-Kang fuzzy system is achieved by a EA, the consequent parameters are estimated by least squares. This reduces the search space dimension leading to greatly reduced load on the EA. The significant contribution of this work is in formulating the fitness function that judiciously applies selection pressure by 1) penalizing low firing strengths of rules, and, 2) by penalizing low rank design matrix at the rule consequents. The proposed method is tested on the identification of non-linear systems.

Keywords: System Identification, Takagi-Sugeno-Kang Fuzzy Systems, Evolutionary Algorithms, Nonlinear Optimization, Data Driven.

1 Introduction

Takagi-Sugeno-Kang fuzzy models are currently the most popular, *model based approach* in modeling uncertainty and non-linearity in controller design problems [2]. Their universal function approximation capability and the *local model* interpretability of the rules make them attractive as design tools for data driven control of non-linear systems [3]. Stability analysis of TSK models is more amenable to mathematical techniques than Mamdani type systems specially if the rule consequents are linear dynamical models. In this paper we concern ourselves with the modeling aspects of the full control problem. In this paper we use TSK model with fixed number of rules and linear dynamic consequents to model a non-linear system. The parametric optimization of the rule base is achieved by a genetic algorithm(GA) with some interesting fitness function choices. This stochastic search based approach is fundamentally different from clustering-based rule extraction schemes [4]. The flexibility of GAs allows for easy incorporation of constraints and prior knowledge that guide the search towards favorable regions of the search space. The designed fitness function allows for system identification with preservation of local character of the fuzzy rules and improved numerical stability for estimation of consequent terms.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 568–576, 2012.

2 Takagi-Sugeno Model

We assume that the system can be described well by n-dimensional state space model driven by m-dimensional control input. Let there be R rules in the fuzzy rule base and each input's universe of discourse is partitioned in to R fuzzy sets. The r^{th} rule of the rule base is described below:

$$\begin{aligned}\text{Rule } r: \text{IF } & x_1 \text{ is } X_{r1} \text{ AND } x_2 \text{ is } \cdots \text{ AND } x_n \text{ is } X_{rn} \text{ AND}\\ & u_1 \text{ is } U_{r1} \text{ AND } u_2 \text{ is } \cdots \text{AND } u_m \text{ is } U_{rm}\\ & \text{THEN } \dot{\mathbf{x}} = \mathbf{o}_r + \mathbf{A}_r\mathbf{x} + \mathbf{B}_r\mathbf{u}\end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{o}_r \in \mathbb{R}^n$, $\mathbf{A}_r \in \mathbb{R}^{n\times n}$ and $\mathbf{B}_r \in \mathbb{R}^{n\times m}$. Here X_{ri} is the fuzzy set corresponding to the r^{th} partition of the i^{th} state variable. U_{rj} is the fuzzy set corresponding to the r^{th} partition of the j^{th} control input.

The truth value (τ) or the firing strength of the r^{th} rule is defined as follows:

$$\tau_r = \prod_{i=1}^{n} \mu_{X_{ri}}(x_i) \prod_{j=1}^{m} \mu_{U_{ri}}(u_j) \tag{1}$$

where we have chosen the product t-norm to infer the truth value of the rule. In the present paper we have used gaussian membership functions defined as:

$$\mu_{X_{ri}}(x_i) = \exp\left(-\frac{(x_i - c_{X_{ri}})^2}{2s^2_{X_{ri}}}\right) \qquad \mu_{U_{rj}}(u_j) = \exp(-\frac{(u_j - c_{U_{rj}})^2}{2s^2_{U_{rj}}}) \tag{2}$$

Normalized truth values (w) are defined $w_r = \frac{\tau_r}{\sum_{r=1}^{R} \tau_r}$ and are known as *fuzzy basis functions* [1] in the literature. The final inferred fuzzy model output is:

$$\dot{\mathbf{x}}_{\text{fuzzy}} = \sum_{r=1}^{R} w_r(\mathbf{o}_r + \mathbf{A}_r\mathbf{x} + \mathbf{B}_r\mathbf{u}) = \sum_{r=1}^{R} w_r \begin{bmatrix}\mathbf{o}_r \ \mathbf{A}_r \ \mathbf{B}_r\end{bmatrix} \begin{bmatrix}1 \ \mathbf{x}^T \ \mathbf{u}^T\end{bmatrix}^T \tag{3}$$

3 System Identification

Parameter estimation involves estimating the values of the parameters of the fuzzy system such that the input-output behavior of the fuzzy system approximates as closely as possible the input-output behavior of the plant. The number of parameters of this fuzzy system is given by $2(n+m)R+(n+n^2+nm)R$ where the first addend corresponds to the premise parameters and the second to the consequent parameters. However the consequent parameters can be estimated using least squares method given the antecedent parameters. Thus a *two step process* can be utilized to estimate all the system parameters. First estimate the antecedent parameters then estimate the consequent parameters and repeat this process until convergence. Assuming that antecedent parameters have been estimated, notice that (3) can be expressed in matrix-vector product form that is *linear in the consequent parameters*:

$$\begin{aligned}\dot{\mathbf{x}}^T &= \left(\left[w_1\ w_2\ \ldots\ w_R\right] \otimes \left[1\ \mathbf{x}^T\ \mathbf{u}^T\right]\right) \left[\mathbf{e}_1\ \mathbf{A}_1\ \mathbf{B}_1\ \mathbf{e}_2\ \mathbf{A}_2\ \mathbf{B}_2 \cdots \mathbf{e}_R\ \mathbf{A}_R\ \mathbf{B}_R\right]^T \\ &= (\mathbf{w}^T \otimes \left[1\ \mathbf{x}^T\ \mathbf{u}^T\right])\mathbf{V} = \mathbf{d}^T\mathbf{V}\end{aligned}$$

where $\otimes$ is the Kronecker product, $\mathbf{w} \in \mathbb{R}^R$ is the vector of normalized truth values and $\mathbf{d} \in \mathbb{R}^{(1+n+m)R}$ is the design vector and $\mathbf{V} \in \mathbb{R}^{(1+n+m)R\times n}$ is the matrix of consequent parameters that needs to be estimated. By stacking as rows the fuzzy model outputs corresponding to each input pattern in the training data one obtains the following system of linear equations in matrix variables:

$$\begin{bmatrix} \vdots \\ \dot{\mathbf{x}_\mathbf{k}}^T \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ (\mathbf{w_k}^T \otimes \left[1\ \mathbf{x_k}^T\ \mathbf{u_k}^T\right]) \\ \vdots \end{bmatrix} \mathbf{V} = \begin{bmatrix} \vdots \\ \mathbf{d_k}^T \\ \vdots \end{bmatrix} \mathbf{V}$$

$$\dot{\mathbf{X}}_{N\times n} = \mathbf{D}_{N\times(1+n+m)R}\mathbf{V}_{(1+n+m)R\times n}$$

We term the matrix $\mathbf{D}$ the *design matrix*, based on the similarity of role of this matrix with the use of design matrices in statistical regression theory. The following Frobenius norm minimization problem is now posed:

$$\min_{\mathbf{V}} \|\mathbf{X} - \mathbf{DV}\|_F^2 \tag{4}$$

We operate on the assumption that the number of data samples(N) is greater than the number of unknown parameters $(nR + n^2R + nRm)$. The first order optimality condition corresponding to (4) yields the *normal equation* given by:

$$\frac{d}{d\mathbf{V}} \mathrm{Tr}((\mathbf{X} - \mathbf{DV})^T(\mathbf{X} - \mathbf{DV})) = \mathbf{0} \Longrightarrow \mathbf{D}^T\mathbf{DV} = \mathbf{D}^T\mathbf{X} \tag{5}$$

However if the *condition number* of $\mathbf{D}$ is large, the condition number of $\mathbf{D}^T\mathbf{D}$ will be worse and the estimate of $\mathbf{V}$ will be inaccurate due to numerical instability. Therefore a least squares solution to the optimization problem (4) is obtained by the use of QR-decomposition method. The consequent parameters are thus estimated by least squares method which reduces the number of unknown parameters from $2(n+m)R + (n+n^2+nm)R$ to $2(n+m)R$, which may be estimated using a GA or derivative based methods.

4 Design of Fitness Function

The fitness function assigns to each genome a fitness value that determines it's reproductive ability. Designing the right fitness function generates a selection pressure towards favorable region of the search space. The primary objective of the fitness function is to improve the approximation of input-output mapping of the plant and the fuzzy model. The input-output mismatch between the two is modeled by the following penalty:

$$\frac{1}{2}\sum_{k=1}^{N}\sum_{i=1}^{n}(\dot{x}_{ki}^{\text{fuzzy}} - \dot{x}_{ki}^{\text{actual}})^2 \tag{6}$$

which measures the approximation error for all the N training patterns. However secondary objectives in learning the TSK model is also to allow for a local model based interpretation of the fuzzy rules and improve the numerical stability properties for consequent terms estimation.

4.1 Penalizing Weakly Firing Rules

The use of this penalty serves to preserve the local character of the fuzzy rules. The firing strength of the rules depends on the following two factors:

1. distance between data samples $\{\mathbf{x_k}, \mathbf{u_k}\}_{k=1}^{N}$ and the centers $\{c_{X_{ri}}, c_{U_{rj}}\}$
2. how localized or diffused the membership functions are i.e. the largeness or smallness of $\{s_{X_{ri}}, s_{U_{rj}}\}$

If the centers are far removed from the data samples and the membership functions are localized i.e. the spread is small then the overall firing strength(ϕ) will be small. This indicates that the fuzzy rule base is not able to capture the essence of the data set. We use the following metric to measure the firing strength of the entire fuzzy rule base with respect to the k-th data sample:

$$\phi_k = \sum_{r=1}^{R_p}\tau_{rk} \quad \text{and} \quad \phi = \sum_{k=1}^{N}\phi_k = \sum_{k=1}^{N}\sum_{r=1}^{R_p}\tau_{rk} \tag{7}$$

ϕ represents the firing strength of the fuzzy rule base with respect to the overall data set. If a large number of ϕ_ks are small then this implies that membership functions do not accurately represent the spread and distribution of the data samples and adjustment in the location and spread of membership functions is required. Thus one of the objectives of the evolutionary optimizer is to ensure adequate firing of the rule base with respect to the overall data set. Based on this idea we propose the use of the following penalty on weak firing:

$$\mathcal{P}_1 = \sum_{k=1}^{N}\mathbf{1}(\phi_k < \epsilon_1)\times\mathcal{N}_1 \quad \text{where,} \quad \mathbf{1}(\phi_k < \text{const}) = \begin{cases} 1, & \text{if } \phi_k < \text{const}, \\ 0, & \text{otherwise.} \end{cases}$$

where $\epsilon_1 \in \mathbb{R}$ is a small real number that lower bounds the firing strength. $\mathcal{N}_1 \in \mathbb{R}$ is a very large number that penalizes the low firing strength of the fuzzy rules. Since the summation over the indicator function serves to count the number of cases which result in low firing of the overall rule base, the penalty $\mathcal{P}_1$ is just an integral multiple of $\mathcal{N}_1$.

4.2 Penalizing Low Rank Design Matrix

Since the consequent parameters are estimated by least squares the condition number and rank of design matrix is very important for this procedure. Thus those genomes which result in higher condition number of the design matrix (cond($\mathbf{D}$)) are penalized. The overall scheme detailing the fitness evaluation of genomes in the GA is given in Algorithm 1. In this paper we used the following values of the parameters $\epsilon_0 = 10^{-15}$, $\mathcal{N}_0 = 10^{15}$, $\epsilon_1 = 10^{-3}$, $\mathcal{N}_1 = 10^9$, $\mathcal{N}_2 = 10^6$ arrived at by some initial simulation runs but these are in general problem dependent though not critically dependent.

Algorithm 1. Fitness Evaluation Scheme of TSK Fuzzy System for use in GA

```
if any $\phi_k < \epsilon_0$ then
  $\mathcal{P}_0 \longleftarrow \sum_{k=1}^{N} \mathbf{1}(\phi_k < \epsilon_0) \times \mathcal{N}_0$ {Penalize Extremely Low Firing}
  return $\mathcal{P}_0$
else
  $\mathcal{P}_1 \longleftarrow \sum_{k=1}^{N} \mathbf{1}(\phi_k < \epsilon_1) \times \mathcal{N}_1$ {Penalize Low Firing}
  if rank($\mathbf{D}$) $\neq (1+n+m)R$ then
    $\mathcal{P}_2 \longleftarrow ((1+n+m)R - \text{rank}(\mathbf{D})) \times \mathcal{N}_2$ {Penalize Low Rank Design Matrices}
    return $\mathcal{P}_1 + \mathcal{P}_2$
  else
    $\mathcal{P}_3 \longleftarrow \frac{1}{2} \sum_{k=1}^{N} \sum_{i=1}^{n} (\dot{x}_{ki}^{\text{fuzzy}} - \dot{x}_{ki}^{\text{actual}})^2$ {Penalize Mismatch Error}
    return $\mathcal{P}_1 + \mathcal{P}_3 + \text{cond}(\mathbf{D})$
  end if
end if
```

5 Results

System identification using the proposed method is demonstrated on two non-linear problems.

5.1 Example 1

The non-linear plant is given by $\dot{x}_1 = x_1 + 2x_2 + u_1$ and $\dot{x}_2 = x_1 - 2x_2^3 + u_2$ for which the universe of discourses are $x_1, x_2, u_1, u_2 \in [-10, 10]$ chosen. MATLAB GA toolbox is used to evolve a population of 100 individuals for 500 generations. Two independent data sets for training and testing each containing 500 randomly generated data points were used. 3 independent runs of the GA were performed and the training and testing error tabulated in Table 1. Run number 2 resulted in the best results. The error in $\dot{x}_1$ is practically zero while the error in $\dot{x}_2$ is dominant. Note however that the range of $\dot{x}_2$ is $[-2000, 2000]$ which is much large compared to the obtained error. Number of rules was fixed to be 3.

Obtained antecedent parameters are tabulated in Table 2 and the consequent parameters are tabulated in Table 3. On examining the results of all the runs we found that the fuzzy sets for linear states or inputs(x_1, u_1, u_2) tend to bunch

Table 1. Training and testing data error for 3 independent runs

Runs	Training Error			Testing Error		
	min	mean	max	min	mean	max
1	0.010461	12.804713	66.237515	0.011640	13.915571	76.176899
2	**0.003482**	**8.623033**	**40.098902**	**0.023402**	**9.175422**	**52.323599**
3	0.027388	9.399573	53.174078	0.006941	10.238333	132.642506

Table 2. Antecedent Parameters centers(c) and spread(s) from Run 2

rule	centers				spread			
	x_1	x_2	u_1	u_2	x_1	x_2	u_1	u_2
1	0.5780	1.7710	3.6593	1.3415	5.7390	5.5089	7.0706	16.2645
2	0.6454	-8.9958	6.2439	-0.1445	5.6781	18.7368	8.9920	11.9889
3	0.7112	2.0586	12.1428	-0.6766	5.6535	6.3080	12.3604	10.2680

Table 3. Consequent Parameters for 3 rule TSK Fuzzy System from Run 2

rule	o	A		B	
1	-1.3731e-17	1.0000e+00	2.0000e+00	1.0000e+00	5.7063e-16
	1.1696e+03	3.4124e+00	1.1673e+02	5.2191e+00	-6.5183e+00
2	-1.7130e-15	1.0000e+00	2.0000e+00	1.0000e+00	-5.0014e-16
	-2.7756e+03	8.9434e-01	-5.2287e+02	-1.9365e-01	-2.8086e-01
3	2.4170e-15	1.0000e+00	2.0000e+00	1.0000e+00	1.1678e-16
	1.6792e+03	-2.1046e+00	1.8351e+02	-4.2989e+00	1.2763e+01

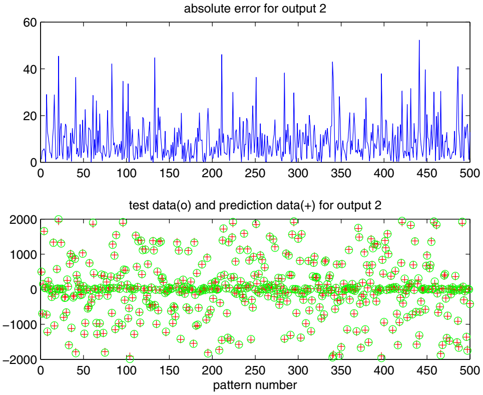


Fig. 1. Absolute error and Predicted Output for $\dot{x}_2$ from Run 2

up together while those of the nonlinear(x_2) do not. As expected the linear optimization method estimates perfectly the consequent term coefficients for the linear inputs. Figure 1 shows the plots of absolute error and prediction accuracy for all the test patterns.

5.2 Example 2

Inverted Pendulum dynamics is given by:

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = \frac{m\ g \sin x_3 \cos x_3 + mlx_4^2 \sin x_3 + fmx_4 \cos x_3 + u}{M + (1\ \ \cos^2 x_3)m}$$

$$\dot{x}_3 = x_4, \quad \dot{x}_4 = \frac{(M+m)(g \sin x_3\ \ fx_4)\ (lmx_4^2 \sin x_3 + u) \cos x_3}{M + (1\ \ \cos^2 x_3)m}$$

where $M = 0.5$, $m = 0.2$, $f = 0.1$, $g = 9.8$ and $l = 0.3$. 3 independent GA runs were performed the results of which are shown in Table 4. Number of rules was fixed at 5. Parameters obtained from run 1 are used. The test error plots for x_2 and x_4 are given in Figure 2. The error plots for x_1 and x_3 are not shown as they are the linear terms. The antecedent parameters obtained are tabulated in Table 5.

Table 4. Training and testing data error for 3 independent runs

Runs	Training Error			Testing Error		
	min	mean	max	min	mean	max
1	**0.000069**	**0.237540**	**1.276169**	**0.000039**	**0.248134**	**1.488931**
2	0.000126	0.242093	0.995847	0.000454	0.259999	1.568845
3	0.002206	0.260516	1.371579	0.001426	0.280484	1.649449

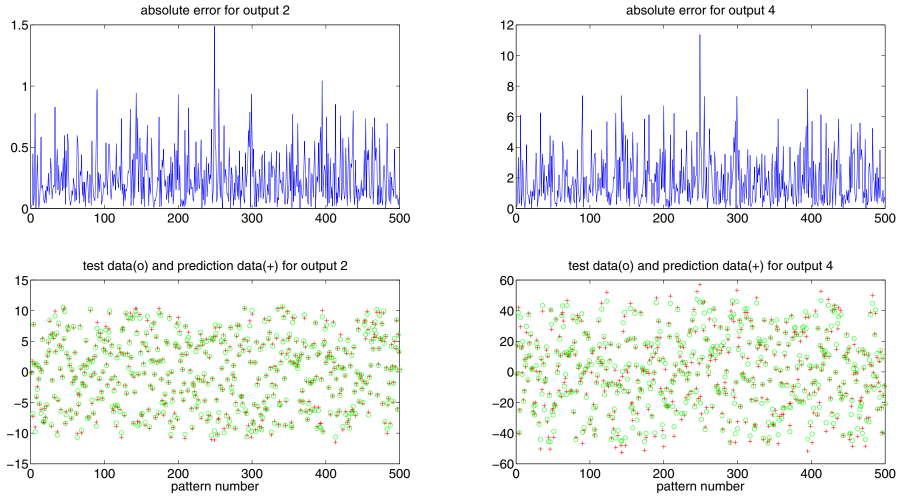


Fig. 2. Absolute error and Predicted Output for $\dot{x}_2$ and $\dot{x}_4$ from Run 1

Table 5. Antecedent Parameters centers(c) and spread(s) from Run 1

rule	centers					spread				
	x_1	x_2	x_3	x_4	u	x_1	x_2	x_3	x_4	u
1	0.7819	1.7048	0.5682	-0.6043	3.1074	0.4819	1.1280	0.7904	2.3669	2.3323
2	0.9549	0.1908	-0.3318	0.8844	3.6432	2.0521	0.4574	0.4847	1.2355	2.1534
3	1.3234	-0.6255	-0.0914	-0.2824	-0.6607	1.7584	1.9120	0.2304	0.3518	2.9835
4	-0.4607	1.4251	-0.4100	0.1244	-3.2593	0.7264	1.3279	-0.5287	0.2590	2.6550
5	-0.2996	-1.7736	0.2699	-0.0415	-2.4419	-0.2333	0.7281	1.1124	2.6875	6.0236

Table 6. Consequent Parameters for 5 rule TSK Fuzzy System from Run 1

rule	e	A				B
1	1.3748e-17	-6.9736e-17	1.0000e+00	1.1238e-16	6.0545e-17	3.0829e-17
	-2.1423e-01	9.3393e-02	1.1134e-01	-2.4186e+00	-4.3801e-02	1.8205e+00
	2.5664e-16	-7.4564e-17	-1.8405e-16	-4.6830e-17	1.0000e+00	-2.3266e-17
	1.6630e+00	-8.7641e-01	-9.1497e-01	3.6490e+01	1.1947e-01	-5.2651e+00
2	2.2482e-16	7 1.2316e-16	1.0000e+00	-2.4221e-17	-1.5832e-17	-9.7632e-17
	-9.2831e-03	2.1943e-01	1.3224e-01	-2.2616e+00	1.6911e-01	1.8797e+00
	9.1552e-17	-7.8456e-17	7.1698e-17	4.9123e-17	1.0000e+00	-4.2189e-17
	3.5770e-01	-1.6846e+00	-9.4497e-01	3.4802e+01	-1.5542e+00	-5.8036e+00
3	-6.9363e-17	1.1350e-16	1.0000e+00	9.8210e-17	2.0358e-18	7.4152e-17
	5.8044e-02	-1.9465e-03	3.7657e-02	-3.5562e+00	9.4262e-03	2.0214e+00
	3.6450e-16	-6.1043e-17	5.6617e-17	1.4731e-16	1.0000e+00	3.5752e-17
	-3.5190e-01	-2.5650e-02	-2.8366e-01	4.3684e+01	7.1944e-02	-6.8461e+00
4	-1.3109e-16	2.4861e-17	1.0000e+00	1.0098e-16	9.1885e-18	-3.6733e-16
	3.8369e-02	-9.3594e-03	-1.1431e-01	-2.5887e+00	-2.1398e-01	1.7867e+00
	-2.1180e-16	5.6893e-17	-1.6697e-16	5.8451e-17	1.0000e+00	-1.2743e-16
	-8.8824e-01	-1.2741e-01	1.1241e+00	3.7611e+01	1.6525e+00	-5.0281e+00
5	-7.7895e-19	1.2846e-16	1.0000e+00	-3.3661e-17	1.2603e-16	5.4574e-17
	9.6398e-02	1.8131e-01	-5.0105e-03	-2.7170e+00	8.5807e-02	1.8661e+00
	9.8415e-17	8.0176e-18	1.3917e-16	3.0206e-17	1.0000e+00	4.3594e-18
	-6.7620e-01	-1.2756e+00	5.3871e-02	3.8480e+01	-7.6091e-01	-5.6068e+00

6 Conclusion

The obtained results demonstrate that effective learning of the system dynamics can be achieved with only a few rules. The absolute error values are small compared to the range of the desired values. Obtained parameters are within the respective universe of discourse reinforcing the local character of the fuzzy rules. While the mean error is acceptable in all of our runs few outliers were also observed. The error for these outliers does not decrease with any increase in number of generations. This happens due to loss of diversity in GA when the entire population converges to a small near optimal region. Hybrid approaches that combine our proposed method and local search techniques for improving the errors due to the outliers is a possible direction for future work.

References

1. Wang, L.-X., Mendel, J.M.: Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. IEEE Transactions on Neural Networks 3(5), 807–814 (1992)
2. Feng, G.: A Survey on Analysis and Design of Model-Based Fuzzy Control Systems. IEEE Transactions on Fuzzy Systems 14(5), 676–697 (2006)
3. Rezaee, B., Zarandi, M.H.F.: Data-driven fuzzy modeling for Takagi Sugeno Kang fuzzy system. Information Sciences 180(2), 241–255 (2010)
4. Babuska, R.: Fuzzy Modeling for Control. Kluwer Academic Publishers, Boston (1998)

A Possibilistic Density Based Clustering for Discovering Clusters of Arbitrary Shapes and Densities in High Dimensional Data

Noha A. Yousri[1,3], Mohamed S. Kamel[2], and Mohamed A. Ismail[1]

[1] Computer and System Engineering, Faculty of Engineering, Alexandria University, Egypt
[2] PAMI, University of Waterloo, Waterloo, Ontario, Canada
[3] Bioinformatics Core, Weill Cornell Medical College, Qatar
noha.yousri@alexu.edu.eg, nay2005@qatar-med.cornell.edu

Abstract. Apart from the interesting problem of finding arbitrary shaped clusters of different densities, some applications further introduce the challenge of finding overlapping clusters in the presence of outliers. Fuzzy and possibilistic clustering approaches have therefore been developed to handle such problem, where possibilistic clustering is able to handle the presence of outliers compared to its fuzzy counterpart. However, current known fuzzy and possibilistic algorithms are still inefficient to use for finding the natural cluster structure. In this work, a novel possibilistic density based clustering approach is introduced, to identify the degrees of typicality of patterns to clusters of arbitrary shapes and densities. Experimental results illustrate the efficiency of the proposed approach compared to related algorithms.

Keywords: Arbitrary Shapes, Arbitrary densities, Possibilistic Clustering.

1 Introduction

Hard or crisp clustering assigns hard memberships to patterns, i.e. a membership is either 0 or 1. This hides a lot of information about the overlapping of clusters, and consequently the correct interpretation of a pattern's belongingness. Fuzzy clustering has overcome this drawback by defining degrees of memberships for patterns with respect to more than one cluster, rather than defining one hard membership for one cluster. This is important in revealing the overlap between different clusters. In biological data, for example, the overlap can reveal the existence of relations between different diseases, and the data patterns that exhibit multi-membership degrees should be investigated for better understanding of such diseases. In medical image analysis, this is also important to reveal the overlap of different tissues in an MRI or an X-ray image. Fuzzy clustering, however is unable to interpret the real relations between patterns and clusters in the presence of noise and outliers. It faces a problem in this case, where each pattern's sum of memberships to all clusters is restricted to 1. Thus, a pattern's membership to one cluster determines its relation to the other clusters. In the presence of outliers and noise, some patterns may not belong to any of the clusters, and the possibility that a pattern belongs to one cluster other than being an outlier is unknown.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 577–584, 2012.

Possibilistic C-Means was introduced in [1] as a new direction to overcome the restricted interpretation of fuzzy clustering. The possibility theory was introduced by Zadeh to deal with another level of uncertainty in the given knowledge. When dealing with clustering, one level of uncertainty is that a pattern might belong to more than one cluster, and that introduced the fuzzy clustering. Another higher level of uncertainty is that a pattern might belong to more than one cluster and might not belong to any. While the possibilistic C-Means [1], solves the above challenge within the framework of the original C-Means, it does not solve the more general clustering problem, where arbitrary shapes and arbitrary density clusters are present in data that contains noise and outliers. Thus it remains a challenge to find an appropriate solution in the domain of non-traditional clustering approaches as density based clustering and others.

This work introduces using a density based measure of relatedness to measure the typicality of a pattern to a cluster. The pattern can show degrees of typicality to different clusters, as well as to being an outlier/noise. It depends on previous density definitions used for known hard density based clustering, as well as new definitions that enable the integration of “possibility” into the main flow of clustering.

2 Related Work

Although Fuzzy C means [2] have been there for ages, the literature on clustering algorithms that tackle the problems of connectedness and arbitrary shaped clusters in a fuzzy approach is still in its infancy. Among the recent approaches that tackled such a problem are: DFC (Density based Fuzzy clustering) presented in [3], F-DBScan (Fuzzy DBScan) of [4] and FLAME (Fuzzy Local Approximation) presented in [5]. FLAME introduces the idea of propagating memberships by linearly combining the fuzzy memberships of neighbour patterns to calculate the membership of a pattern. The drawback of the algorithm is the huge number of different cluster cores that can be initially found, which present the number of initial clusters. This increases the time complexity, and dealing with clusters of arbitrary shapes is also not clear. The work presented here is independent from the above mentioned algorithms, defining novel measures for possibilistic clustering, which is able to give a wider interpretation of the relations between patterns and clusters.

3 Proposed Approach

A Local Intra-cluster Density/Distance Attraction (LIDA) membership is introduced. This measure brings forward a deeper analytic approach that can reveal the relation of a pattern’s neighbourhood density characteristics to the inner most dense parts of a cluster, thus measuring a degree of typicality to the cluster. Using this approach, one can easily explore the patterns at various levels of cluster memberships in a single cluster. Based on this measure, a clustering algorithm is proposed that is able to find out the genuine clusters of arbitrary shapes, and to discover clusters of different densities in the same dataset.

Density Relatedness: The concept of density relatedness between patterns can be used to avoid static models of density based clustering, where users define a static density threshold. The static model may lead to undesired results if clusters' densities widely vary. On the other hand, presenting a threshold that guides the choice of ***relatively*** denser patterns is more appropriate. In that case, different clusters in the same dataset can have different or arbitrary densities. This overcomes the drawbacks of currently known density based approaches, and presents a solution to a wider range of data sets, which is more important to high dimensional data applications.

LIDA Approach

The importance of the LIDA approach is depicted in discriminating between more dense and less dense patterns in the same cluster. It gives possibilistic degrees based on the relative density of patterns to their neighbouring patterns' densities' characteristics.

The cluster's inner structure is reflected by the Intra-Cluster Density Attraction membership (LIDA) defined as follows:

Definition: *Local Intra-cluster Density Attraction LIDA:* Let *p* a pattern in dataset *P*, and $NN_{\varepsilon}(p)$ the ε-neighborhood of p, defined as (see [6]):

$$NN_{\varepsilon}(p) = \{q \in P \mid d(p,q) \leq \varepsilon\} \tag{1}$$

then the LIDA membership of pattern p to a cluster c is measured as:

$$\mu_{p,c} = \frac{|NN_{\varepsilon}(p)|}{\max\limits_{S1}\{|NN_{\varepsilon}(q)|\}} .\mu_{x,c} \tag{2}$$

where S1 is determined as follows:

$$S1 = \{q \in NN_{\varepsilon}(p) \cap c \; : \; |NN_{\varepsilon}(q)| > |NN_{\varepsilon}(p)| \wedge \frac{|NN_{\varepsilon}(p)|}{|NN_{\varepsilon}(q)|} .\mu_{\mathbf{q,c}} > thresh\}$$

$$x = pattern(\max_{S1}\{|NN_{\varepsilon}(q)|\})$$

where *pattern*(.) is a function that maps the criteria $\max\limits_{S1}\{|NN_{\varepsilon}(q)|\}$ to a pattern id.

The value of the LIDA membership determines how related a pattern is to the densest parts of a cluster. A pattern's eligibility to join a cluster c is defined by S1 above, which examines the neighbours of p that belong to that cluster, i.e. $q \in NN_{\varepsilon}(p) \cap c$, and at the same time such neighbors should have denser neighbourhoods than p, i.e. $|NN_{\varepsilon}(q)| > |NN_{\varepsilon}(p)|$. Given those two conditions, pattern p joins the cluster if, given a value for *thresh,* the following condition is satisfied:

$$\frac{|NN_{\varepsilon}(p)|}{|NN_{\varepsilon}(q)|}.\mu_{\mathrm{q,c}} > thresh$$

This condition imposes the relatedness of pattern p to pattern q with respect to density, and weighed by the membership of q to its cluster, to decide whether q can transfer the cluster membership to p (or in other words : p can join the cluster through q). Thus, it is a recursive relationship that propagates the density of the inner most dense patterns to the outer less dense patterns.

A pattern can join a number of clusters in its neighbourhood, thus merging up those clusters into one cluster. The final membership for pattern p to its cluster is then determined as stated in (2).

A clustering algorithm (see figure 1) is based on this measure that joins patterns according to the degree of their LIDA membership. A threshold *thresh* is used to determine if a membership allows a pattern to join the cluster. To be able to calculate the memberships appropriately, the most dense patterns are visited first. Thus sorting

```
    Algorithm: LIDA Possibilistic Clustering
Input: data set P, $\varepsilon$, thresh
Output: Set of Clusters C, Patterns possibilistic memberships
 Begin
    T←Construct a metric tree for P
    For each pattern p in P
         { $NN_{\varepsilon}(p)$ ←Retrieve neighbours in range $\varepsilon$ from p, using T}
     L←Sort patterns ascendingly on neighborhood size
    For each pattern p in L
         For each q in $NN_{\varepsilon}(p)$
                  If $|NN_{\varepsilon}(p)| > |NN_{\varepsilon}(q)|$ and $\frac{|NN_{\varepsilon}(p)|}{|NN_{\varepsilon}(q)|}.\mu_{\mathbf{q,c(q)}} > thresh$
                           If p is singleton {c(q)=c(q) $\cup$ p}   //merge it to q's cluster
                              Else {Merge (c(q), c(p))}   //merge two clusters
                           End If
              End If
         End For
         If |c(p)|>1      {Calculate $\mu_{p,c(p)}$ }  //pattern joined a cluster
            Else {Create a new cluster c(p)={p}, $\mu_{p,c(p)}$=1} //p singelton
          End If
    End For
    C={}       //remove outlier clusters and get set of genuine clusters
    For all clusters c {if $|c| > 0.01.|P|$ { $C = C \cup c$ }}
End
```

Fig. 1. LIDA possibilistic clustering

the patterns on their density is an initial requirement, followed by scanning the patterns in order of their density. Patterns having denser neighbourhoods compared to their neighbours' neighborhoods are the first patterns in their clusters, taking a membership value of 1. The final set of clusters is determined by all clusters of a significant size, where a constant threshold on the size (more than 1% of the total dataset size) is used in all experiments.

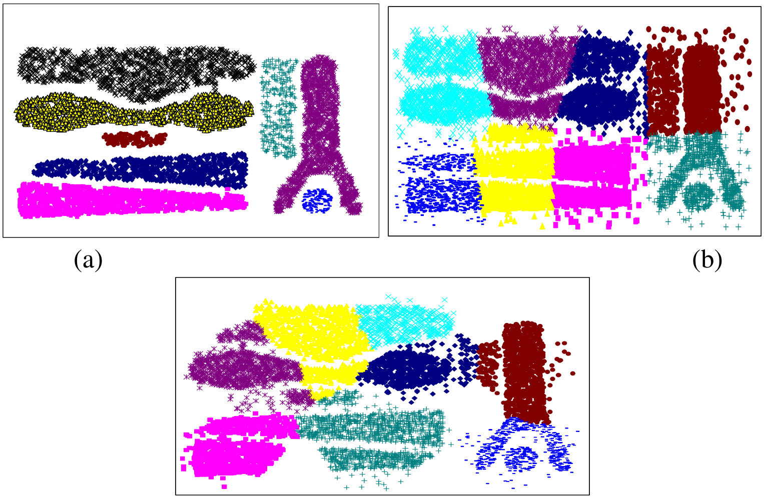
As shown in figure 1, the algorithm takes two parameters ε, and *thresh* as input. The first determines the size of neighbourhood, and the second determines the degree of acceptance of a pattern into a cluster according to their density relation. As the value of *thresh* decreases, more patterns can be accepted into the cluster, and more clusters can be merged together Whereas increasing this value constrains the addition of patterns of less dense neighborhoods to much denser clusters. At the same time, increasing this value can result in switching some patterns from merging with denser clusters, with other patterns that are more related to them, even if they have loose neighborhoods, which is an expected outcome of a density relatedness concept. The algorithm's complexity reaches an average of *O(n.d.log(n))*- where n is the number of patterns and d is the dimensionality- when considering a binary metric tree structure. For building a binary metric tree, it takes the same time complexity.

Parameter tuning can be done using a validity measure as proposed in [9], after thresholding the membership values. However, for dealing with possibilistic memberships, a suitable validity should be developed in the future and used to tune the parameters.

4 Experimental Results

To illustrate the efficiency of the proposed approach, a 2-D data set is used for visualizing the results, and another two high dimensional sets are used to examine the competence of the proposed approaches to FCM and the recent algorithm of [5], available online as the Gedas software (simple Flame was used, with tuning only the k parameter, and leaving other parameters as default). Both the efficiency of LIDA as a clustering algorithm and as a possibilistic clustering approach are examined. A 2-D dataset (DS5 -8000 patterns) that is used by [7] and [8] to illustrate their efficiency is used. The data set is used as an example of a set having arbitrary shaped clusters with arbitrary densities**.** Only efficient algorithms such as Chameleon and Mitosis can obtain such a clustering. Other algorithms as DBScan [6] fail to find the genuine clusters in such a data set. Thus it is used to illustrate the efficiency of using the density relatedness model of LIDA to find clusters of arbitrary densities. Figure 2.a shows the results of LIDA, using ε=10 and *thresh*=0.35, which corresponds to the clustering obtained by Chameleon and Mitosis. Note different symbols (and colors) used for clusters determine their uniqueness. The results for FCM (at C=8) are illustrated in figure 2.b, and that for Flame/Gedas (at k=350, the only parameter which gave 8 clusters among other surrounding values explored, and Euclidean distance selected) is shown in figure 2.c. It is shown how FCM results in finding globular clusters rather than finding the natural cluster shapes. That would also be expected from the possibilistic C-means which depends on the means being centers of clusters, restricting cluster finding to globular shaped ones. Similarly the results obtained by

Flame/Gedas show a globular shaped clusters, rather than finding the true clusters. It is also important to note that it was shown before - in [7],[8]- that algorithms as DBScan and SNN are unable to get clusters of arbitrary densities because of their dependency on a relatively static density based model (with DBScan more strict than

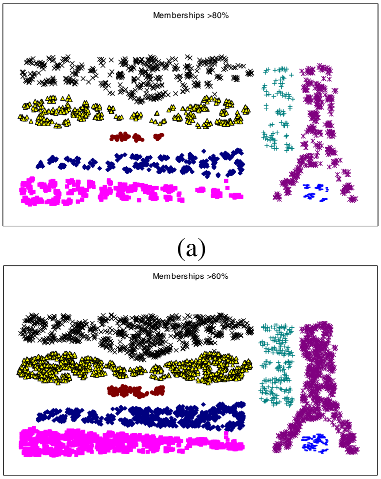


(a)

(b)

(c)

Fig. 2. (a) Results of LIDA on Chameleon's data DS5, at ε =10, thresh=0.35, (b) Results for FCM, at C=8, (c) Results for simple Flame (Gedas software), at k=350 (8 clusters). Each color represents a cluster.



(a)

(b)

Fig. 3. Results of LIDA at ε =10, thresh=0.35, showing patterns of (a) possibilistic memberships > 0.8 and (b) possibilistic memberships>0.6

SNN). Results in figures 3.a and 3.b show patterns which are associated with higher possibilistic LIDA memberships.

The comparison is also done for the Synthetic Control Charts set –SCC - (see UCI repository) of 600 patterns and 60 dimensions and 6 clusters. This data is labeled, and thus external validity indexes (Jaccard, Rand, and Adjusted Rand) are used for comparison. Figure 4 shows that LIDA results for SCC at ε =0.75 and thresh=0.25 are more valid compared to those obtained by FCM (at C=6, which is the original number of classes) and those obtained by Flame/Gedas (at k=40, which gave the best validity index measure among other explored values :10,15,30,50,60, with Pearson Correlation selected). Higher results for the used validity indexes indicate better clustering results. There is no trimming of patterns at a selected membership, however around 100 patterns are left as singletons (belong to no clusters) in the LIDA results, and patterns are allocated to 3 main clusters (compared to Mitosis [8] that obtained 4 clusters). Whereas, as known for the FCM, all patterns are clustered to one of the possible 6 clusters. For the Flame/gedas, k=40 obtained 3 clusters and 6 outliers, indicating that even with the higher number of outliers obtained by LIDA, LIDA results show higher validity values. It is thus shown how LIDA outperforms both algorithms.

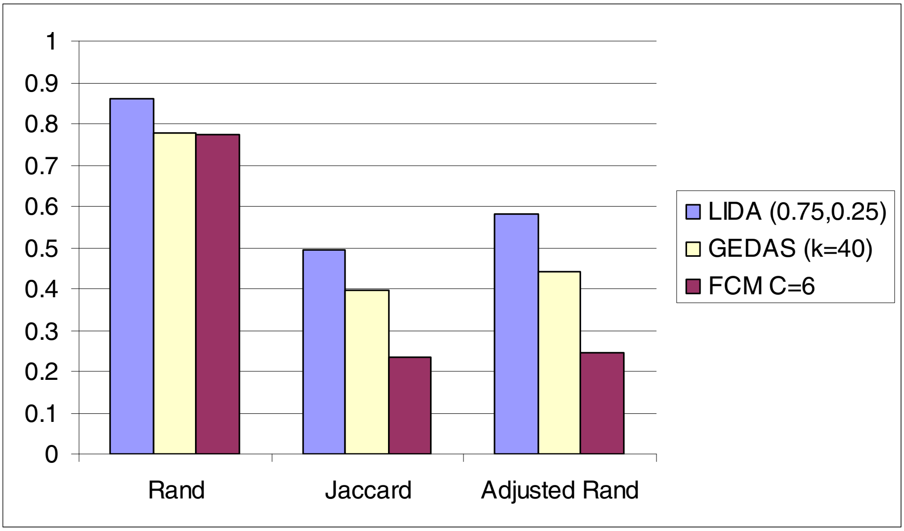


Fig. 4. Comparing LIDA to Flame/Gedas (Simple Flame) and FCM on Synthetic Control Charts set using external validity Indexes

Another high dimensional dataset, the leukemia dataset (999 genes and 38 samples (dimensions) with two clusters, was used to compare LIDA to FCM. An internal validity is used [9] to evaluate clusters in this case, for absence of class labels. Results obtained by LIDA are more valid than those obtained by FCM when the number of genes that pass a certain fuzzy or possibilistic memberships are around 400. At ε =0.5 and thresh=0.25, LIDA obtains a validity of 0.0015 (after trimming genes at possibilistic memberships higher than 15% or 25%, resulting in clusters of 375 and 429 genes). Whereas FCM obtains a validity of 0.0026 (after trimming at fuzzy membership 0.75 to give 405 genes).The minimum value indicates more valid clusters. For this dataset, the trimming at specific fuzzy memberships was required for a fair comparison, thus Flame/Gedas was hard to compare to, since the fuzzy membership values are generally not included in the clustering result output –only the final clusters after thresholding are obtained.

5 Conclusion

In this work, a novel possibilistic clustering approach is introduced to find the degree of a pattern's typicality to a cluster in the presence of outliers. The algorithm is designed to be able to find clusters of arbitrary shapes and densities in a possibilistic framework. The experimental results on 2-D and higher dimensional sets illustrate its performance compared to a recent developed algorithm (Flame using Gedas software) and the most commonly used uncertainity based clustering Fuzzy C-Means.

References:

1. Krishnapuram, R., Keller, J.M.: A Possibilistic Approach to Clustering. IEEE Transactions on Fuzzy Systems (1993)
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algoritms. Plenum Press, New York (1981)
3. Baltunas, L., Gordevicius, J., Halkidi, M., Vazirgiannis, M.: DFC: A Density-based Fuzzy Clustering Algorithm. In: Panhellenic Conference on Informatics, PCI (2005)
4. Kriegel, H.P., Pfeifle, M.: Density Based Clustering of Uncertain Data. In: KDD (2005)
5. Fu, L., Medico, E.: FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. BMC Bioinformatics 8(3) (2007)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial data sets with noise. In: KDD (1996)
7. Karypis, G., Han, E.H., Kumar, V.: Chameleon: A hierarchical clustering algorithm using dynamic modeling. Computer 32(8), 68–75 (1999)
8. Yousri, N.A., Kamel, M.S., Ismail, M.A.: A distance-relatedness dynamic model for clustering high dimensional data of arbitrary shapes and densities. Pattern Recognition (2009)
9. Yousri, N.A., Kamel, M.S., Ismail, M.A.: A Novel Validity Measure for Clusters of Arbitrary Shapes and Densities. In: ICPR (International Conference of Pattern Recognition) (2008)

A Knowledge-Driven Bi-clustering Method for Mining Noisy Datasets

Karima Mouhoubi, Lucas Létocart, and Céline Rouveirol

LIPN, Université Paris 13, Sorbonne Paris Cité, CNRS (UMR 7030), France
{karima.mouhoubi,Lucas.letocart,celine.rouveirol}@lipn.univ-paris13.fr

Abstract. Bicluster discovery is an important task in various experimental domains. We propose here a new biclustering system *COBIC*, which combines graph algorithms with data mining methods to efficiently extract highly relevant and potentially overlapping biclusters. *COBIC* is based on maximum flow / minimum cut algorithms and is able to take into account background knowledge expressed as a classification, by a weight adaptation mechanism when iteratively extracting dense regions. The proposed approach, when compared on three real datasets (Yeast gene expression datasets) with recent and efficient biclustering algorithms shows very good performances.

Keywords: Constrained biclustering, noisy datasets, dense subgraphs, maximal flow/minimal cut.

1 Introduction

Itemset mining in boolean data consists in finding all rectangles of 1 in a boolean matrix. When such a boolean matrix comes from numerical data resulting from complex experimental processes, it may contain noise. One effect of the noise when mining itemsets is to shatter relevant itemsets satisfying some constraints (such as the minimal support constraint) into an exponential number of small irrelevant fragments. Many approaches have been proposed [21,4,16,6] in order to take into account the effect of noise in itemset mining that introduce a *density* constraint on the patterns to be mined. A large majority of the proposed methods use the level-wise principle of Apriori algorithm [1]. An exception is the work of Poernomo and colleagues [19] in which a constraint of noise proportional to the support of the itemset is considered.

We have proposed in [17] a system *HANCIM* and shown empirically that *HANCIM* heuristically builds a relatively small number of regions that have comparable or better quality (both in terms of supervised or unsupervised criteria) than state of the art systems constructing (bi)-clusters from noisy datasets. We propose here an extension of *HANCIM* that guides the extraction of dense regions with background knowledge, expressed as a classification. This guidance takes place by adapting weights when building the bipartite graphs that support the computation of dense regions by maximum flow / minimum cut algorithms. This guidance is soft : it is used to adapt the weights in the bipartite graphs to

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 585–593, 2012.

reward coherence with respect to a potentially sparse classification. It smoothly generalizes must-link constraints from constrained clustering approaches [23] and extends it to a biclustering context. It can also be seen as some kind of semi-supervision of biclustering [2]. The rest of the paper is organised as follows: section 2 sketches a state of the art in the area itemset mining in noisy contexts and in particular, about extraction of overlapping biclusters. Section 3 describes our original contribution in this paper. In section 4, we describe how our approach is evaluated and compared to state of the art biclustering algorithms and conclude in section 5.

2 Related Work

We consider in the following a finite set of attributes A, a finite set of observations O, and a binary relationship $R \subseteq O \times A$. R can be modelled by a boolean matrix. An *itemset* m is a subset of A. Many approaches have been proposed in order to handle noise during itemset mining. These approaches can be divided into *complete* and *heuristic* methods. Complete (discrete) methods look for all itemsets that satisfy a set of constraints, including a density and a min support constraint, hopefully both anti-monotonic. As a consequence, they proceed with an APRIORI level-wise search [1] and prune their search space according to their anti-monotonic constraints [21,16,4]. Let us mention other studies [22,15] that have focused on the enumeration of all maximal bicliques that verify a minimum density constraint, called *quasi-bicliques*, in transactional data. However, these methods are very costly in execution time and provide a high number of (redundant) results.

To overcome the limitations of *complete* methods, especially when working on large and noisy datasets, non-complete or *heuristic* methods have been applied, such as biclustering approaches [7,20,9,11]. In contrast to classical clustering techniques, biclustering does not require attributes in the same cluster to behave similarly over all the observations. Instead, a bicluster is defined as a subset of attributes that exhibit compatible values over a subset of observations.

CC [7] uses a greedy search heuristic to generate arbitrarily positioned, overlapping co-clusters, based on a homogeneity constraint. Their algorithm is expensive and it identifies individual coclusters sequentially, which may quickly deteriorate the quality of obtained biclusters. The *Plaid model* approach [13] improves upon this by directly modeling overlapping clusters, but still cannot identify multiple co-clusters simultaneously. The *BiMax* system [20] proposes a methodology for comparing and validating biclustering methods that handle a binary reference model. It proposes a simple divide-and-conquer combinatorial algorithm that exactly determines all optimal and maximal groupings, and produces a number of co-clusters exponential in the number of genes and experiments, making it impractical in case of large datasets. *OPSM* [3] looks for submatrices in which the expression levels of all the genes induce the same linear ordering of the experiments. *BiMax* and *OPSM* outperforms other biclustering approaches. However, *BiMax* on one hand produces a huge amount of results, among which a high number are irrelevant and on the other hand, *OPSM*, although very accurate, is designed to identify only a single co-cluster. The *ROCC* system [9], like *CC*, generates arbitrarily positioned, overlapping biclusters, using a two-step approach. The method is quite sophisticated but requires a high number of parameters to be set before learning. *SScorr* [18] uses an evolutionnary technique relying on a fitness function based on the linear correlation among genes to search for potentially overlapping bicluters. This approach also requires setting a high number of parameters. *Bagged Biclustering* [11] is another recent method for generating potentially overlapping biclusters, the main limitation is that it requires a priori setting the number of searched biclusters K.

3 Proposed Approach

Our goal is to efficiently build a relatively small number of maximally dense biclusters without any specification on the number of biclusters or on their size. Our previous algorithm *HANCIM* iterates a two-step approach : it first identifies a bicluster (O_0, A_0) with density 100% and then uses the attribute set A_0 named also *seed pattern* s to find, in a second step, a dense bicluster (O_j, A_j) such that $s \subseteq A_j$. The bicluster (O_j, A_j) should satisfy two constraints : i) each attribute of A_j has a density in O_j greater than or equal to a threshold δ, and ii) each observation of O_j is strongly associated to each attribute of A_j (see [17]).

In order to mine a maximal dense bicluster which includes a given seed pattern $s \in 2^A$, a bipartite valuated graph associated to s is first built, and then a minimum cut is computed. We adapt the capacities assigned to edges so as to recover, after computing a minimum cut, a dense subgraph which includes the attributes of the seed s and the set of observations O_j that are strongly associated to these attributes. At the next step, the graph corresponding to the observations set O_j is constructed so as to recover, after computing a minimum cut, a subset of attributes that have densities greater than or equal to δ for the observations in O_j. These two steps are alternatively repeated on observations and attributes until the dense subgraphs extracted at steps l and $l+1$ are identical, in this case our subgraph can not be extended anymore and the process is stopped.

3.1 Constrained Biclustering

We present in this section our methodology, named *COBIC* (COnstrained BI-Clustering), for extracting relevant biclusters from noisy contexts. Our goal in this work is to exploit background information in the form of a reference classification to guide the extraction of dense regions. This guidance takes place by adapting weights when building the bipartite graphs that support the computation of dense regions by maximum flow / minimum cut algorithms. This guidance is soft as it is used to adapt the weights in the bipartite graphs handled to implement a light coherence with respect to a potentially sparse classification C_Y.

Definition 1. *Let Y be a finite set of elements (attributes or observations). We define a classification of Y as $C_Y = \{C_i \; / \; C_i \subseteq Y\}$.*

No constraint is imposed on the structure of C_Y, which can be sparse (some elements of Y may not belong to any class in C_Y or belong to several classes), and may not be a partition (given two C_i and C_j, $i \neq j$ of C_Y it may be the case that $C_i \cap C_j \neq \emptyset$).

3.2 Adapting Weights

We detail in Algorithm 1 the construction of a weighted graph for a set of elements (attributes or observations) denoted by X_l at the l^{th} step. The purpose is to extract a set of attributes satisfying the constraint of minimum density after computing the minimum cut. When $l > 2$, we compute a quality measure for both extracted element sets Y_{l-1} and Y_{l-2} in the last two iterations $l-1$ and $l-2$. The quality of a set Y_i is evaluated in terms of its similarity with all classes $C_i \in C_Y$ defined on Y. At each step, our objective is to guide the search of biclusters in order to favour the computation of sets of elements coherent with classes of a classification C_Y, and not with a single class of C_Y.

Given the behaviour of our algorithm in the last two iterations, we check the similarity of sets Y_{l-2} and Y_{l-1} with classes in C_Y, denoted $sim_{Y_{l-2}}$ and $sim_{Y_{l-1}}$ in the following. When the similarity of set Y_{l-1} extracted during the previous iteration $l-1$ is better than the similarity of Y_{l-2}, the algorithm performs well and so we construct the graph corresponding to X_l as we done in *HANCIM* (line 24 of algorithm 1). Otherwise (line 9), we use the similarity values to weight the capacities of the edges of the graph. In fact, if the similarity of Y_{l-1} is less than the one of Y_{l-2}, this means that our solution moves away from classification C_Y which is either due to elements of Y_{l-2} removed from Y_{l-1} or/and new elements added to Y_{l-1}. In this case, at step l, by weighting the capacities of edges incident to vertices y_j belonging to $Y_{l-1} \setminus Y_{l-2}$ by the value of the similarity $sim_{Y_{l-1}}$, we penalize these vertices y_j, and by weighting the capacities of edges incident to vertices y_k belonging to Y_{l-2} by the value of the similarity $sim_{Y_{l-2}} > sim_{Y_{l-1}}$, we indicate our preference towards these vertices y_k compared to y_j. To do this, we weight the capacities of the edges (x_i, y_j) such that $y_j \in Y_{l-1} \setminus Y_{l-2}$ by the value of similarity $sim_{Y_{l-1}}$ and the edges (x_i, y_j) such that $y_k \in Y_{l-2}$ by the value of similarity $sim_{Y_{l-2}}$. Knowing that the capacities differ according to the graph construction associated to attributes or to observations, the original capacities, proposed in [17] are defined as follows :

1. if $X_l \subseteq O$:
   - $W_{x_i y_j}=\dfrac{100}{|X_l|}$ (lines 14, 17, 19 of algorithm 1) and
   - $W_{y_j t}=2 \times (100 \times \delta)$ - $weight^{-}(y_j)$ (line 22 of algorithm 1).
2. if $X_l \subseteq A$:
   - $W_{x_i y_j}=(\dfrac{d^{+}(x_i)}{max_{x_k \in X_l}(d^{+}(x_k))} + \dfrac{d^{-}(y_j)}{max_{y_k \in Y_j}(d^{-}(y_k))})$ x $\dfrac{100}{|X_l|}$ (lines 14, 17, 19 of algorithm 1) and
   - $W_{y_j t}= max_{y_k \in Y_j}(d^{-}(y_k)) \times \frac{200}{|X_l|}$ - $weight^{-}(y_j)$ (line 22 of algorithm 1).

As $sim_{Y_{l-1}} < sim_{Y_{l-2}}$, by weighting the capacities of the edges $(x_i,\, y_j)$ such that $y_j \in Y_{l-1} \setminus Y_{l-2}$ with the similarity value $sim_{Y_{l-1}}$, the capacities of these edges $(x_i,\, y_j)$ is greatly reduced and thus the possibilty to cut these edges and so to suppress vertices $y_j \in Y_{l-1} \setminus Y_{l-2}$ from Y_l is increased.

Algorithm 1. CONSTRUCT_GRAPH_COBIC

```
input  : $D = (O, A)$: Dataset, $X_l$: vertices set ($X_l \subseteq A$ or $X_l \subseteq O$) and $l > 2$, $\delta$: density
         threshold, $C$: Classification of $Y$ (if $X_l \subseteq A$ then $Y = O$, else $Y = A$ ), $Y_{l-1} \subseteq Y$:
         vertices extracted at step $l-1$, $Y_{l-2} \subseteq Y$: vertices extracted at step $l-2$
output: $G(V, E)$: the graph constructed
begin
    $sim_{Y_{l-1}}$ = SIMILARITY($Y$, $C_Y$, $Y_{l-1}$);
    $sim_{Y_{l-2}}$ = SIMILARITY($Y$, $C_Y$, $Y_{l-2}$);
    if $(sim_{Y_{l-1}} < sim_{Y_{l-2}})$ then
        $V = X_l \cup \{s, t\}$ ;
        forall the $x_i \in X_l$ do
            $E = E \cup (s, x_i)$;
            weight$(s, x_i) = +\infty$ ;
        forall the $x_i \in X_l$ do
            forall the $y_j$ s.t. $D[x_i][y_j] == 1$ do
                $V = V \cup y_j$ ;
                $E = E \cup (x_i, y_j)$ ;
                if ( $y_j \in Y_{l-2}$) then
                    weight$(x_i, y_j) = W_{x_i y_j} \times sim_{Y_{l-2}}$ ;
                else
                    if ( $y_j \in Y_{l-1}$) then
                        weight$(x_i, y_j) = W_{x_i y_j} \times sim_{Y_{l-1}}$ ;
                    else
                        weight$(x_i, y_j) = W_{x_i y_j}$ ;
        forall the $y_j \in V \setminus (X_l \cup \{s, t\})$ do
            $E = E \cup (y_j, t)$ ;
            weight $(y_j, t) = W_{y_j t}$;
    else
        CONSTRUCT_GRAPH_HANCIM (cf [17]);
```

For calculating the average similarity between a set of attributes (resp. observations) and some classes of attributes (resp. observations), we consider Y a set of elements (attributes or observations), C_Y a classification of Y and Sub_Y a subset of Y. We compute our similarity between $Sub_Y \subset Y$ and $C_i \in C_Y$ as the Jaccard similarity between the partitions $(Sub_Y, Y \setminus Sub_Y)$ and $(C_i, Y \setminus C_i)$. So, Jaccard similarity is defined as follows :
$Sim\_JACCARD(Y, Sub_Y, C_i) = \dfrac{N_{01} + N_{10}}{N_{01} + N_{01} + N_{11}}$, where $N_{11} = |Sub_Y \cap C_i|$, $N_{01} = |C_i \setminus Sub_Y|$, $N_{10} = |Sub_Y \setminus C_i|$ and $N_{00} = |(Y \setminus Sub_Y) \cap (Y \setminus C_i)|$.

As mentioned previously, we use several classes of C_Y to guide our search, so we compute an average value over k best similarities. We have empirically set k to 5 in our experiments.

4 Experiments and Results

All computational tests were run on a Linux PC with an Intel(R) Pentium(R) 4 (3 GHz) microprocessor and 2GB of RAM. Evaluation of *HANCIM* with respect to artifical datasets have been performed in previous work [17]. The advantage of evaluation on artificial datasets is that true classes are available, allowing the implementation of supervised measures. In this work, we chose to evaluate the performance of *COBIC* on two yeast microarray datasets, the Gasch dataset [10] and the Lee dataset [14], due to the availability of a reference classification, correct but expected to be highly incomplete, to guide the biclustering process. The Gasch dataset consists of the expression values of 6152 yeast genes under 173 environmental stress conditions. The Lee dataset consists of gene expression values of 5612 yeast genes across 592 experiments. To assess the biological relevance of the biclusters extracted from these real datasets, we rely on the enrichment of extracted biclusters in GO terms, as done in [5], according to the p-value (lower p-values denote highly significant associations) contained in public databases such as Gene Ontology (GO) [8] and Kyoto Encyclopedia of Genes and Genomes (KEGG) [12].

We compare the performance of *COBIC* with prominent biclustering algorithms, i.e., *HANCIM* [17], *SScorr* [18] and *ROCC* [9]. Through extensive experimentations, Prelic et al. [20] have already shown that *OPSM* [3] and *BiMax* [20] outperform other previous and well known biclustering algorithms, More recently, Deodhar et al. [9] have shown that *ROCC* outperforms *OPSM* and *BiMax*, and Mouhoubi et al. [17] shown that *HANCIM* outperforms *BiMax* both on synthetic and real datasets. As in [9], in order to compare *COBIC* with *ROCC* and *SScorr*, we select our 200 best results (with the best p-values). To compare our results with those obtained by *BiMax*, for both *COBIC* and *HANCIM*, we use the same discretization model described in [20] and a discretization threshold set to $e_{min} + (e_{max} - e_{min})/2$ where e_{min} and e_{max} represent the minimum and maximum gene expression values in the data context. We ran *COBIC* and *HANCIM* on the discretized Gasch and Lee datasets with minimum support set of 20% and density threshold of 80%. We use KEGG for the weight adaption phase in *COBIC*. Evaluation is made with respect to GO terms' enrichment of biclusters for both approaches. As GO contains more results than KEGG (GO contains 4227 clusters of size 14 in average, as opposed to 99 clusters of size 37 in average for KEGG), this means that sparse but correct information is enough to improve results from *HANCIM*. Computation time are rather small except on the Lee dataset : 3min and 35min for the restricted and complete Gasch datasets and 280min for the Lee dataset. Indeed, on the Lee dataset, we obtain almost twice the number of results as for the Gasch dataset, the average number of iterations to converge for each result is more important and each iteration of *COBIC* is also more time-consuming, as the Lee dataset contains 3.5 times more genes than the Gasch dataset. The additional time induced by the weight learning phase is limited to approximately 20% of the total computation time.

Gasch dataset. First, we compare *COBIC* with *HANCIM* on the restricted Gasch dataset used in [20,17]. Table 1 gives the percentage of GO-term's enrichment of computed biclusters for each method, in which at least one GO term is over-represented for different levels of significance.

As we can see in table 1, the best results are obtained by *COBIC* over *HANCIM* ; the number of results is almost equivalent (6 more with *COBIC*), but the proportion of biclusters significantly enriched by a GO term of the biological process hierarchy for *COBIC* is always greater than for *HANCIM* for all levels of significance. We observe the same behaviour if we compare *COBIC* with the results of *SScorr* in [18], but here *COBIC* is better than *SScorr* by a larger amount. On the restricted Gasch dataset, for the 100 best results of *SScorr*, the percentage of enriched biclusters with a p-value less than 0.01 (resp. 0.001) is less than 30%, (resp. 20%). If we take our 200 best biclusters (see table 1), 100% of our enriched biclusters have a p-value less than 0.001. Table 1 gives also the percentage of GO-term enriched biclusters for *COBIC* in which one or several GO terms are over-represented for different levels of significance on the complete Gasch dataset. As we can see, results obtained by *COBIC* on the complete Gasch dataset are better than on the restricted Gasch dataset. All p-values associated to the 200 best biclusters obtained with *COBIC* are inferior to e^{-10} and the best p-value is e^{-98}.

Lee dataset. The third column of table 1 gives the same information for the results obtained from the Lee dataset. We can see that all p-values associated to the 200 best biclusters obtained with *COBIC* are inferior to e^{-03} and with best p-values, $5.46e^{-43}$ and $1.58e^{-34}$ correspond to the GO processes that were already indentified in the Gasch dataset. Note that the percentage of the gene (attributes) represented in at least one bicluster in the 200 best results of *ROCC* is 16.5% for the Lee dataset, whereas 30% of the genes are represented in the 200 best results of *COBIC*.

Table 1. Obtained results on restricted and complete Gasch dataset and Lee dataset

Dataset	restricted Gasch			complete Gasch		Lee	
	HANCIM	*COBIC*	*COBIC* 200 best	*COBIC*	*COBIC* 200 best	*COBIC*	*COBIC* 200 best
# results	548	554	200	1075	200	1969	200
$<e^{-2}$	94%	97.3%	100%	96%	100%	86%	100%
$<e^{-3}$	48%	58%	100%	60%	100%	34.5%	100%
$<e^{-4}$	28%	33.7%	93.5%	41.5%	100%	11%	77%
$<e^{-5}$	18%	24%	66.5%	35.6%	100%	6%	40%
$<e^{-10}$	7%	10%	28%	25%	100%	1.5%	8.5%
$<e^{-20}$	4%	6.5%	18%	15%	83%	0.2%	2%
Best p-value	e^{-38}	e^{-64}	e^{-64}	e^{-98}	e^{-98}	e^{-43}	e^{-43}

5 Conclusion

In this paper, we have evaluated an original biclustering technique, COBIC, taking into account background knowledge expressed as a classification. A comparison with other methods demonstrates that COBIC is both efficient and quite competitive. This opens new possibilities for mining heterogeneous multi-view datasets. Indeed, while in this paper we focused on biclustering microarray datasets, it would be worthwhile to study the applicability of adapted or generalized instances of *COBIC* to different application domains, like community extraction in social networks.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Proc. SIGMOD, pp. 207–216. ACM Press (1993)
2. Basu, S., Banerjee, A., Mooney, R.: Semi-supervised clustering by seeding. In: Proc. ICML 2002, pp. 27–34 (2002)
3. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. In: Proc. RECOMB, pp. 49–57 (2002)
4. Besson, J., Robardet, C., Boulicaut, J.F.: Mining a new fault-tolerant pattern type as an alternative to formal concept discovery. In: Proc. ICCS, pp. 144–157 (2006)
5. Birmele, E., Elati, M., Rouveirol, C., Ambroise, C.: Identification of functional modules based on transcriptional regulation structure. BMC 2(Suppl. 4), S4 (2008)
6. Cheng, H., Yu, P.S., Han, J.: Approximate frequent itemset mining in the presence of random noise. Soft Comp. Kno. Dis. Data Min., 363–389 (2008)
7. Cheng, Y., Church, G.: Biclustering of expression data. In: ISMB, pp. 8:93–103 (2000)
8. Cherry, J.M., Adler, C., Ball, C., Chervitz, S.A., Dwight, S.S.: SGD: Saccharomyces genome database. Nucleic Acids Research 26(1), 73–79 (1998)
9. Deodhar, M., Gupta, G., Ghosh, J., Cho, H., Dhillon, I.S.: A scalable framework for discovering coherent co-clusters in noisy data. In: Proc. ICML 2009, p. 31 (2009)
10. Gasch, A., Spellman, P., Kao, C., Carmel-Harel, O., Eisen, M., Storz, G., Botstein, D., Brown, P.: Genomic expression programs in the response of yeast cells to environmental changes. Mol. Biol. Cell 11(12), 4241–4257 (2000)
11. Hanczar, B., Nadif, M.: Using the bagging approach for biclustering of gene expression data. Neurocomputing 74(10), 1595–1605 (2011)
12. Kanehisa, M., Goto, S.: KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res. 28(1), 27–30 (2000)
13. Lazzeroni, L., Owen, A.: Plaid models for gene expression data. Statistica Sinica 12, 61–86 (2000)
14. Lee, I., Date, S., Adai, A., Marcotte, E.: A probalistic functionnal network of yeast genes. Science 306(5701), 1555–1558 (2004)
15. Li, J., Sim, K., Liu, G., Wong, L.: Maximal quasi-bicliques with balanced noise tolerance: Concepts and co-clustering applications. In: SDM, pp. 72–83 (2008)
16. Liu, J., Paulsen, S., Sun, X., Wang, W., Nobel, A.B., Prins, J.: Mining approximate frequent itemsets in the presence of noise: Algorithm and analysis. In: SDM (2006)
17. Mouhoubi, K., Létocart, L., Rouveirol, C.: Itemset mining in noisy contexts: A hybrid approach. In: Proc. ICTAI 2011, pp. 33–40 (2011)
18. Nepomuceno, J., Lora, A.T., Aguilar-Ruiz, J.: Biclustering of gene expression data by correlation-based scatter search. BioData Mining 4(3) (2011)
19. Poernomo, A.K., Gopalkrishnan, V.: Towards efficient mining of proportional fault-tolerant frequent itemsets. In: Proc. KDD 2009, pp. 697–706 (2009)

20. Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9), 1122–1129 (2006)
21. Seppänen, J.K., Mannila, H.: Dense itemsets. In: Proc. KDD 2004, pp. 683–688 (2004)
22. Uno, T., Arimura, H.: Ambiguous Frequent Itemset Mining and Polynomial Delay Enumeration. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 357–368. Springer, Heidelberg (2008)
23. Wagstaff, K., Cardie, C.: Clustering with instance-level constraints. In: Proc. ICML 2000, pp. 1103–1110 (2000)

Robust Active Learning for Linear Regression via Density Power Divergence

Yasuhiro Sogawa[1], Tsuyoshi Ueno[2],
Yoshinobu Kawahara[1], and Takashi Washio[1]

[1] ISIR, Osaka University, 8-1, Mihogaoka, Ibaraki, Osaka, Japan
[2] Japan Science and Technology Agency, 1-4-14, Shibata, Kita-ku, Osaka, Japan
sogawa@ar.sanken.osaka-u.ac.jp, tsuyoshi.ueno@gmail.com,
{kawahara,washio}@ar.sanken.osaka-u.ac.jp

Abstract. The performance of active learning (AL) is crucially influenced by the existence of outliers in input samples. In this paper, we propose a robust pool-based AL measure based on the density power divergence. It is known that the density power divergence can be accurately estimated even under the existence of outliers within data. We further derive an AL scheme based on an asymptotic statistical analysis on the M-estimator. The performance of the proposed framework is investigated empirically using artificial and real-world data.

Keywords: Active Learning, Density Power Divergence, Regression.

1 Introduction

Recent development of information technology has made it possible to collect huge amount of data automatically in various domains. In most cases, such data are composed of majority unlabeled-instances and minority labeled-instances. This is because labeling tasks by human experts or additional experiments (oracles) are usually expensive or time-consuming. For example, in a car insurance company, an insurance fee is determined by its company's employees based on car information, driver's driving records and so on. However, such determination by hand needs enormous cost and time. In recent years, active learning (AL) has been discussed to make learning processes with majority unlabeled-instances and minority labeled-instances more efficient [1]. In contrast to passive learning, AL selects some unlabeled instances expected to be informative for learning and asks an user to label them. This AL framework has been widely applied successfully in various regions, such as speech recognition [2] and classification [3].

One of the most important problems in AL is how to select unlabeled instances called queries and several querying measures have been discussed over the last few decades [4,5]. These conventional AL methods commonly assume that oracles always give correct labels on instances. In the real-world, however, human experts might give incorrect labels due to their conditions or additional experiments might make mistakes due to their environments. Such an oracle giving noisy labels is called a noisy oracle. With a noisy oracle, the accuracy of

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 594–602, 2012.

model estimation by AL could become worse. Thus, in this paper, we propose a new AL algorithm to tackle this problem caused by a noisy oracle.

Among various types of query measures, in this paper, we employ Variance Reduction Approach (VRA) [6], which is based on an estimation variance of parameters (estimators). In this approach, AL algorithms select queries that are expected to minimize the estimation variance and aim to derive better parameters. A conventional method based on VRA use Kullback-Leibler (KL) divergence in estimating model parameters and the Fisher information criterion in determining queries [7]. However, the KL-divergence-based methods do not consider noisy-oracles and thus work worse if there are noisy labels. Therefore, in this paper, we employ robust divergences called density power divergence, which are known to be robust measures for evaluating the difference between two distributions. Through the asymptotic analysis on M-estimator, we incorporate the density power divergence into our querying measure based on VRA.

The remainder of the paper is organized as follows. In Section 2, we first briefly review the pool-based AL framework and the conventional AL method based on VRA. In Section 3, we extend VRA through an asymptotic analysis on M-estimator and apply it to the density power divergence. Then, in Section 4, we propose a practical querying measure based on the discussion in the previous section. Finally, we show experimental results using artificial and real-world datasets in Section 5 and conclude our paper in Section 6.

2 Preliminaries

2.1 Pool-Based Active Learning

Let us consider the pool-based AL, which is a frequently-discussed framework for situations where the distribution of input instances is unknown but instances from the true input distribution are given [3]. Formally, in pool-based AL framework, it is assumed that one has a small set of labeled instances $\mathcal{L} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_{n_l}, y_{n_l})\}$ and a large set of unlabeled instances $\mathcal{U} = \{\mathbf{x}_{n_l+1}, \cdots, \mathbf{x}_{n_u}\}$ $(n_l \ll n_u)$. Then, one tries to find a set of queries from $\mathcal{U}$ that is expected to be informative for estimating a 'good' model. An overall procedure of the pool-based AL algorithm is described in Algorithm 1. At the beginning of the algorithm, a model with parameter $\boldsymbol{\theta}$ is estimated from the small set of labeled instances $\mathcal{L}$ (Algorithm 1, Step 1).

Next, the algorithm select the most 'informative' subset of unlabeled instances as queries (Algorithm 1, Step 2a). Then, each query is labeled by an oracle and added to $\mathcal{L}$ as labeled instances (Algorithm 1, Step 2b, c). These learning and querying steps are repeated iteratively.

As mentioned above, the selection of an informativeness measure for queries is an important problem in developing pool-based AL algorithms. One of the promising measures is based on the estimation variance which evaluates an efficiency of an estimator $\boldsymbol{\theta}$. The strategy for minimizing this estimation variance is known as VRA [6]. However, a conventional method, which will be explained in the next subsection, does not consider mis-labeled instances.

Algorithm 1. Pool-based active learning algorithm

```
– Input
    • $\mathcal{L}$: Set of labeled instances
    • $\mathcal{U}$: Set of unlabeled instances
    • $K$:Number of queries per an iteration
    • $T$:Number of querying iteration
– Main
    1. $\boldsymbol{\theta}^{(0)}$ = LearnModel($\mathcal{L}$)
    2. for $i$=1,$\cdots$, $T$
        (a) $\mathcal{S}$ = SelectQuery($\mathcal{U}$,$K$,$\boldsymbol{\theta}^{(i-1)}$)
        (b) $\mathcal{S}_{labeled}$ = AddLabel($\mathcal{S}$)
        (c) $\mathcal{L} = \mathcal{L} \cup \mathcal{S}_{labeled}$, $\mathcal{U} = \mathcal{U} \backslash \mathcal{S}$
        (d) $\boldsymbol{\theta}^{(i)}$ = LearnModel($\mathcal{L}$)
    3. end
– Output
    • $\boldsymbol{\theta}^{(T)}$: Estimated parameters
```

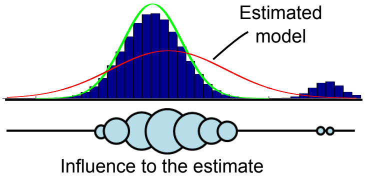


Fig. 1. An illustration of the weighted estimator

2.2 A Conventional Method Based on VRA

In this subsection, we review the conventional AL method based on VRA [7]. This AL method estimates a model under an assumption that the model $p_{\boldsymbol{\theta}}(\mathbf{x}, y)$ with a parameter $\boldsymbol{\theta}$ includes a true distribution $q(\mathbf{x}, y)$, i.e. $q(\mathbf{x}, y) = p_{\boldsymbol{\theta}^*}(\mathbf{x}, y)$ where $\boldsymbol{\theta}^*$ is a true parameter. The model parameter $\boldsymbol{\theta}$ is obtained by minimizing the KL-divergence. Generally, given n labeled instances, the model parameter $\boldsymbol{\theta}$ is obtained by solving the following equation:

$$\sum_{i=1}^{n} \partial_{\boldsymbol{\theta}} \log p(\mathbf{x}_i, y_i; \hat{\boldsymbol{\theta}}_n) = 0, \tag{1}$$

where $\partial_{\boldsymbol{\theta}}$ denotes the partial derivation with respect to $\boldsymbol{\theta}$. The parameter $\hat{\boldsymbol{\theta}}_n$ estimated by solving Eq. (1) is called a maximum likelihood estimator (MLE) and is known to converge to $\boldsymbol{\theta}^*$ if $n \to \infty$. Then, the method uses the variance of the parameter, $\mathbb{E}_q[(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)^\top]$, as the querying measure, where $\mathbb{E}_q[\cdot]$ is the expectation over a set of $\{\mathbf{x}, y\}$ with respect to $q(\mathbf{x}, y)$, and selects queries to minimize the difference between $\hat{\boldsymbol{\theta}}_n$ and $\boldsymbol{\theta}^*$. This measure is called an estimation variance and corresponds to the Fisher information. Refer [7] for the detail. If outliers exist, this method tends to overfit and therefore to behave worse.

3 Querying Measure by Asymptotic Analysis

This section presents a querying measure in the proposed framework. We extend the conventional VRA scheme to be applicable to consist estimators based on the various divergence by using a general class of consistent estimators, so-called *M-estimators*. In Section 3.1, we show the notion of the M-estimators and their statistical aspects, which are basis of our querying measure. In Section 3.2, we introduce robust estimators based on the density power divergence, and propose new querying measures which provide us with the robust queries in the noisy oracle situations.

3.1 Asymptotic Analysis on M-estimator

The M-estimator is a general class of consistent estimators including the MLE. The advantage of such the general class of estimators is to know common statistical aspects among various estimators without focusing on a particular estimator. Suppose we have i.i.d. labeled instances $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ generated from a distribution $q(\mathbf{x}, y) = p_{\boldsymbol{\theta}^*}(\mathbf{x}, y)$. A function $\boldsymbol{\psi}(\mathbf{x}, y; \boldsymbol{\theta})$ is called an *estimating function* when it satisfies the conditions for any $\boldsymbol{\theta}$:

$$\mathbb{E}_{\boldsymbol{\theta}}\left[\boldsymbol{\psi}(\mathbf{x}, y; \boldsymbol{\theta})\right] = \mathbf{0}, \tag{2}$$

where $\mathbb{E}_{\boldsymbol{\theta}}[\cdot]$ and $\det|\cdot|$ denote the expectation with respect to $p_{\boldsymbol{\theta}}(\mathbf{x}, y)$ and a determinant of the matrix, respectively. If the estimating function exists, an estimator $\hat{\boldsymbol{\theta}}_n$, which possesses desirable asymptotic properties, is obtained by solving the following estimating equation:

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\mathbf{x}_i, y_i; \hat{\boldsymbol{\theta}}_n) = \mathbf{0}. \tag{3}$$

A solution of Eq. (3) is called an M-estimator in statistics. The following proposition states a convergence of the M-estimator, $\hat{\boldsymbol{\theta}}_n \to \boldsymbol{\theta}^*$ if $n \to \infty$ (consistency) and its asymptotic estimation variance.

Proposition 1. *Suppose we have i.i.d. labeled instances* $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ *generated from a distribution* $q(\mathbf{x}, y)$ *and a function* $\boldsymbol{\psi}(\mathbf{x}, y; \boldsymbol{\theta})$ *satisfies the condition* (2)*. Then, if* $n \to \infty$*, the M-estimator* $\hat{\boldsymbol{\theta}}_n$ *converges to* $\boldsymbol{\theta}^*$ *in probability. Moreover,*

$$\mathbb{E}_q[(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)^\top] = \frac{1}{n}\mathbf{A}_q^{-1}\mathbf{M}_q(\mathbf{A}_q^{-1})^\top, \tag{4}$$

where

$$\mathbf{A}_q = \mathbb{E}_q\left[\partial_{\boldsymbol{\theta}}\boldsymbol{\psi}(\mathbf{x}, y; \boldsymbol{\theta}^*)\right], \ \mathbf{M}_q = \mathbb{E}_q[\boldsymbol{\psi}(\mathbf{x}, y; \boldsymbol{\theta}^*)\boldsymbol{\psi}(\mathbf{x}, y; \boldsymbol{\theta}^*)^\top]. \tag{5}$$

The proof of this proposition is given in [8]. The results in Proposition 1 allow us to generalize the conventional VRA scheme so as to utilize not only the MLE, but also any M-estimators.

3.2 Density Power Divergence

The weakness of the KL-based VRA is that overfitting often occurs in the estimation by this method if outliers exist. And, the querying measure based on overfitted parameters might give noisy queries. To alleviate this weakness of the KL-based VRA, we incorporate robust divergences into VRA. For such divergences, we focus on the density power divergences, particularly β-divergence and γ-divergence. The estimators based on these divergences are known as M-estimators.

The density power divergence is a class of statistical measures between two probabilistic distributions, and has been developed to realize robust estimation against unanticipated outliers. One in the class is called the β-divergence proposed in [9]. The divergence between $q(\mathbf{x}, y)$ and $p_{\boldsymbol{\theta}}(\mathbf{x}, y)$ is defined by

$$D_\beta(q\|p_{\boldsymbol{\theta}})=\frac{1}{(1+\beta)}\left\{\frac{1}{\beta}\iint q(\mathbf{x}, y)^{1+\beta}\mathrm{d}\mathbf{x}\mathrm{d}y - \iint q(\mathbf{x}, y)p_{\boldsymbol{\theta}}(\mathbf{x}, y)^{\beta}\mathrm{d}\mathbf{x}\mathrm{d}y + \iint p_{\boldsymbol{\theta}}(\mathbf{x}, y)^{1+\beta}\mathrm{d}\mathbf{x}\mathrm{d}y\right\},$$

where β is a positive constant. Note that the β-divergence converges to the KL-divergence if $\beta \to 0$. Therefore, this can be regarded as a generalization of the KL-divergence. Estimation based on the β-divergence can be achieved through the minimization of this divergence. The β-divergence estimator is given as a solution of the following estimating equation:

$$\sum_{i=1}^{n} \boldsymbol{\psi}_\beta(\mathbf{x}_i, y_i; \hat{\boldsymbol{\theta}}_n) = \sum_{i=1}^{n} \Big(p_{\hat{\boldsymbol{\theta}}_n}(\mathbf{x}_i, y_i)^{\beta} \partial_{\boldsymbol{\theta}} \ln p_{\hat{\boldsymbol{\theta}}_n}(\mathbf{x}_i, y_i) - \iint p_{\hat{\boldsymbol{\theta}}_n}(\mathbf{x}, y)^{\beta+1} \partial_{\boldsymbol{\theta}} \ln p_{\hat{\boldsymbol{\theta}}_n}(\mathbf{x}, y) \mathrm{d}\mathbf{x}\mathrm{d}y \Big) = \mathbf{0}, \quad (6)$$

which is derived by taking the partial derivative of the β-divergence and by replacing the expectation with respect to $q(\mathbf{x}, y)$ with its sample mean. Note that since $\boldsymbol{\psi}_\beta(\mathbf{x}, y; \boldsymbol{\theta})$ satisfies the condition (2), the estimator obtained from Eq. (6) is an M-estimator. Thus, the β-divergence estimator shares the same solution with the MLE.

The common property of all density power divergence is to take the self-weighted log-likelihood estimating equation, such as Eq. (6). These weighted estimating equations allow us to estimate parameters in disregard of outliers. Fig. 1 demonstrates how the density-power-divergence-based estimator reduces the influence of outliers. Since outliers generally tend to have lower probabilities with repeat to model $p_{\boldsymbol{\theta}}$, the weights on outliers automatically become small. This characterizes the density power divergence as a robust estimator.

The γ-divergence, a variant of the β-divergence, is defined as follows [10]:

$$D_\gamma(q \| p_{\boldsymbol{\theta}}) = \frac{1}{\gamma+1} \left\{ \frac{1}{\gamma} \ln \iint q(\mathbf{x}, y)^{1+\gamma} \mathrm{d}\mathbf{x}\mathrm{d}y - \ln \iint q(\mathbf{x}, y) p_{\boldsymbol{\theta}}(\mathbf{x}, y)^{\gamma} \mathrm{d}\mathbf{x}\mathrm{d}y + \ln \iint p_{\boldsymbol{\theta}}(\mathbf{x}, y)^{1+\gamma} \mathrm{d}\mathbf{x}\mathrm{d}y \right\},$$

where γ is a positive constant. The γ-divergence also converges to the KL-divergence if $\gamma \to 0$. Also, the estimate of the γ-divergence is given by:

$$\sum_{i=1}^{n} \boldsymbol{\psi}_\gamma(\mathbf{x}_i, y_i; \hat{\boldsymbol{\theta}}_n) = \sum_{i=1}^{n} \left(\frac{p_{\hat{\boldsymbol{\theta}}_n}(\mathbf{x}_i, y_i)^{\gamma} \partial_{\boldsymbol{\theta}} \ln p_{\hat{\boldsymbol{\theta}}_n}(\mathbf{x}_i, y_i)}{\sum_{i=1}^{n} p_{\hat{\boldsymbol{\theta}}_n}(\mathbf{x}_i, y_i)^{\gamma}} - \frac{\iint p_{\hat{\boldsymbol{\theta}}_n}(\mathbf{x}, y)^{\gamma+1} \partial_{\boldsymbol{\theta}} \ln p_{\hat{\boldsymbol{\theta}}_n}(\mathbf{x}, y) \mathrm{d}\mathbf{x}\mathrm{d}y}{\iint p_{\hat{\boldsymbol{\theta}}_n}(\mathbf{x}, y)^{\gamma+1} \mathrm{d}\mathbf{x}\mathrm{d}y} \right) = \mathbf{0}.$$

This can be regarded as the normalized version of Eq. (6). Similar to the β-divergence estimator, the γ-divergence estimator obtained from the above equation is also an M-estimator since $\boldsymbol{\psi}_\gamma(\mathbf{x}, y; \boldsymbol{\theta})$ satisfies the condition (2).

As a result, we can obtain the estimating variance based on the β- and γ-divergence using the estimating functions $\boldsymbol{\psi}_\beta(\mathbf{x}, y; \boldsymbol{\theta})$ and $\boldsymbol{\psi}_\gamma(\mathbf{x}, y; \boldsymbol{\theta})$ for the M-estimators.

4 Empirical Measures for Querying

Based on the asymptotic variance of Eq. (4), the estimating functions $\boldsymbol{\psi}_\beta(\mathbf{x}, y; \boldsymbol{\theta})$ and $\boldsymbol{\psi}_\gamma(\mathbf{x}, y; \boldsymbol{\theta})$ in the previous section, we explain empirical querying measures in our AL methods. For simplicity, we collectively denote the estimating functions by $\boldsymbol{\psi}(\mathbf{x}, y; \boldsymbol{\theta})$. Particularly, in this paper, we discuss the following linear regression model with Gaussian noise: $p(y|\mathbf{x}; \mathbf{w}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mathbf{w}^T\mathbf{x})^2}{2\sigma^2}\right)$

where $\mathbf{w}$ is a coefficient parameter of the regression model and σ is a standard deviation. For simplicity, we denote the collection of these parameters by $\boldsymbol{\theta}$.

Our strategy of AL is to minimize the estimation variance in Eq. (4). However, since the true distribution $q(\mathbf{x}, y) = p_{\boldsymbol{\theta}^*}(\mathbf{x}, y)$ is not known, Eq. (5) cannot be calculated directly. Thus, similar to the existing work [7], we calculate the estimators of $\mathbf{A}_q$ and $\mathbf{M}_q$ as in the following, using the parameter $\hat{\boldsymbol{\theta}}_n$ estimated from a set of labeled instances and the sample mean of query instances:

$$\widehat{\mathbf{A}}_{p_{\hat{\boldsymbol{\theta}}_n}}(\mathcal{S}) = \sum_{\mathbf{x}_i \in \mathcal{S}} \int p_{\hat{\boldsymbol{\theta}}_n}(y|\mathbf{x}_i) \partial_{\boldsymbol{\theta}} \boldsymbol{\psi}(y|\mathbf{x}_i; \hat{\boldsymbol{\theta}}_n) \mathrm{d}y, \tag{7}$$

$$\widehat{\mathbf{M}}_{p_{\hat{\boldsymbol{\theta}}_n}}(\mathcal{S}) = \sum_{\mathbf{x}_i \in \mathcal{S}} \int p_{\hat{\boldsymbol{\theta}}_n}(y|\mathbf{x}_i) \boldsymbol{\psi}(y|\mathbf{x}_i; \hat{\boldsymbol{\theta}}_n) \boldsymbol{\psi}(y|\mathbf{x}_i; \hat{\boldsymbol{\theta}}_n)^{\top} \mathrm{d}y. \tag{8}$$

If a model consists of an unique parameter, the above equations are scalars. In this case, the estimation variance of the parameter given as a product of them is obtained by selecting a set of queries $\mathcal{S}$ to minimize the variance. However, in our case, the model has more than two parameters and the estimation variance needs to be optimized over a matrix. Therefore, we take the trace norm of the matrix and derive the querying measure as follows:

$$\mathcal{S}^* = \mathop{\arg\min}_{\mathcal{S} \subseteq \mathcal{U} \wedge |\mathcal{S}| = K} \frac{1}{2K} \mathrm{tr} \left\{ \widehat{\mathbf{A}}_{p_{\hat{\boldsymbol{\theta}}_n}}(\mathcal{S})^{-1} \widehat{\mathbf{M}}_{p_{\hat{\boldsymbol{\theta}}_n}}(\mathcal{S}) (\widehat{\mathbf{A}}_{p_{\hat{\boldsymbol{\theta}}_n}}(\mathcal{S})^{-1})^{\top} \right\}. \tag{9}$$

The procedure of taking a trace norm is known as *A-optimality* and is popular in AL [11]. However, the querying measure based on the γ-divergence still cannot be calculated directly due to the integration in it. Therefore, we employ the Monte Carlo integration to calculate it. The optimization of $\mathcal{S}$ is known as a combinatorial problem that is difficult to be solved. Thus, we utilize the greedy algorithm to determine the set $\mathcal{S}$ similar to conventional AL algorithms [6].

5 Experiments

In this section, we show some experimental results to illustrate the performance of our AL method using artificial and real-world datasets. In these experiments, we compared the following six methods: three standard random-query algorithms based on the KL-, β- and γ-divergence (**KL-, $\boldsymbol{\beta}$- and $\boldsymbol{\gamma}$-RAND**); a conventional AL algorithm based on the KL-divergence ([7] applied in linear regression) (**KL-AL**); our proposed AL methods based on the β- and γ-divergence (**$\boldsymbol{\beta}$- and $\boldsymbol{\gamma}$-AL**). As for KL-AL and KL-RAND, parameters were estimated by solving Eq. (3) analytically. And, as for the other methods, we employ quasi-Newton's method for estimating parameters. Moreover, in these experiments, the sample size for the Monte Carlo integration in γ-AL was set to be 250 and the parameter values β and γ were set to be 0.1.

5.1 Demonstration with Artificial Datasets

In the first experiment, we investigated the robustness of the proposed methods using artificial datasets. The procedure for generating the datasets is as follows:

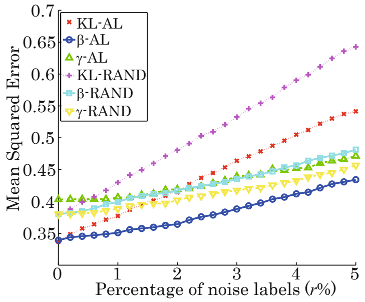


Fig. 2. Difference among the five methods in increasing noisy labels

Table 1. Characteristics of Datasets

Dataset	# of Dim.	# of Instances
concrete	8	1030
machine	7	209
elevator	7	9517

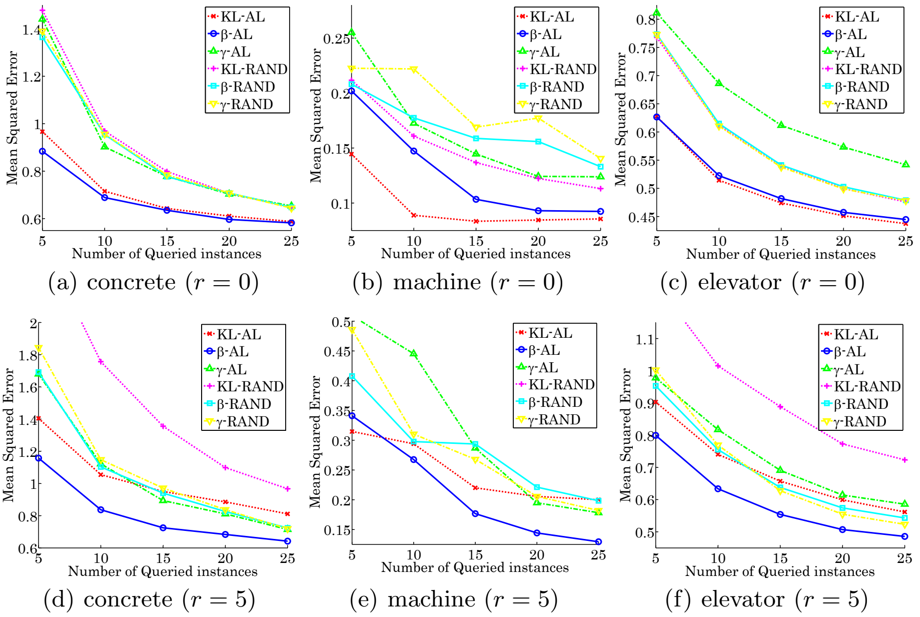
First, we randomly generated instances $\mathbf{x}_i$ from a uniform distribution in the range of [-1, 1], where the dimensionality and the number of instances are respectively 5 and 300. Next, we randomly generated five-dimensional coefficient vector $\mathbf{w}$ from a uniform distribution in the range of [-2.5, 2.5]. Moreover, noises e_i in linear regression models were randomly generated from a Gaussian distribution with zero mean and unit variance. Finally, we determined labels y_i as $y_i = \mathbf{w}^\top \mathbf{x}_i + e_i$.

Each of the generated datasets is randomly partitioned into the training set with 80% instances and the test set with 20% instances. 10 instances in the training set were randomly selected as initial labeled instances $\mathcal{L}$. Then, noises ±5 are added to $r\%(r = 0, 0.2, \cdots, 5)$ of randomly selected labels in the remaining instances $\mathcal{U}$. In this experiment, the number of iterations for querying T was set to be 2 and the number of queries K is set to be 5.

Fig. 2 shows the means-squared error (MSE) between true and estimated labels on test instances by the methods. The values in the graph are averaged over 2000 random trials for numerical stabilization. As can be seen in Fig. 2, with the increasing of noisy labels, the average errors by the KL-divergence-based methods grow more rapidly than the β/γ-divergence-based methods. Also, in most cases, AL methods seem to perform better than the random-query methods. However, the performance of γ-AL was worse than β-AL and the other random-query methods. This would be because the querying measure of γ-AL selects less informative instances due to an approximation by the Monte Carlo integration. Although one could improve the approximation by increasing the number of samples, it usually leads severe increase of computational costs. Thus, these results seem to show that β-AL is more feasible.

5.2 Experiments with Real-World Datasets

Next, we conducted experiments with 3 real-world datasets provided from [12,13]. The summaries of the datasets are given in Table 1. First, each of the datasets is partitioned into an initial set $\mathcal{L}$, an unlabeled set $\mathcal{U}$ and a test set in the same



(a) concrete ($r = 0$) (b) machine ($r = 0$) (c) elevator ($r = 0$)

(d) concrete ($r = 5$) (e) machine ($r = 5$) (f) elevator ($r = 5$)

Fig. 3. Comparisons of means-squared errors among six methods at each learning step

manner with the previous experiment. Then, noises ±5 were added to labels of $r\%(r = 0, 5)$ instances in $\mathcal{U}$ as noisy labels. For this experiment, 300 instances were subsampled from $\mathcal{U}$ as candidates for unlabeled instances before selecting queries if the cardinality of $\mathcal{U}$ is more than 300. In this experiment, we set the number of learning iteration T and queries K to 5, respectively. Similar to the previous experiment, we evaluated the average MSE of 1000 random trials.

The graphs in Fig. 3 show the errors at each learning step of the methods. As can be seen in Fig. 3, the error of β-AL is comparable with KL-AL without noisy labels. On the other hand, in cases where noisy labels exist, the errors of KL-AL become larger than β-AL. Similar to the result of the previous experiment, γ-AL seems to work worse than the other methods in most cases. Thus, our proposed method β-AL seems to work better than the other methods in this experiment.

6 Summary

We proposed the robust AL methods by incorporating density power divergence into VRA. Our querying measures were obtained by the asymptotic analysis on the M-estimator including the β- and γ-divergence estimator. The proposed methods can achieve robust results under situations with a noisy oracle due to the properties of the robust divergences. We investigated the performance of our methods by the experiments with the artificial datasets and the real-world datasets and confirmed that it could achieve desirable performance levels even under the situation with a noisy oracle. To show the practicality of our AL strategy, we just applied it to a linear regression model in this paper. Our strategy, however, is not restricted to the model and therefore one of our future works is to apply our scheme into other models and investigate their behaviors.

References

1. Campbell, C., Cristianini, N., Smola, A.: Query learning with large margin classifiers. In: Proceedings of the 17th Int. Conf. on Machine Learning, pp. 111–118 (2000)
2. Hakkani-Tur, D., Riccardi, G., Gorin, A.: Active learning for automatic speech recognition. In: IEEE Int. Conf. on Acoustics, Speech and Signal Processing, vol. 4, pp. 3904–3907 (2002)
3. McCallum, A., Nigam, K.: Employing EM in pool-based active learning for text classification. In: Proceedings of the 15th Int. Conf. on Machine Learning, pp. 350–358 (1998)
4. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
5. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the Conf. on Empirical Methods in Natural Language Processing, pp. 1070–1079. Association for Computational Linguistics (2008)
6. Settles, B.: Active learning literature survey. Technical Report Computer Science Technical Report 1648, University of Wisconsin-Madison (2010)
7. Zhang, T., Oles, F.: The value of unlabeled data for classification problems. In: Proceedings of the 17th Int. Conf. on Machine Learning, pp. 1191–1198 (2000)
8. Van der Vaart, A.: Asymptotic statistics. Cambridge Univ. Pr. (2000)
9. Basu, A., Harris, I., Hjort, N., Jones, M.: Robust and efficient estimation by minimising a density power divergence. Biometrika 85(3), 549–559 (1998)
10. Fujisawa, H., Eguchi, S.: Robust parameter estimation with a small bias against heavy contamination. Journal of Multivariate Analysis 99(9), 2053–2081 (2008)
11. Hoi, S., Jin, R., Zhu, J., Lyu, M.: Batch mode active learning and its application to medical image classification. In: Proceedings of the 23rd Int. Conf. on Machine Learning, pp. 417–424 (2006)
12. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
13. Torgo, L.: Regression datasets, `http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html`

Adaptive Probabilistic Policy Reuse

Yann Chevaleyre[1] and Aydano Machado Pamponet[2]

[1] Université Paris 13
yann.chevaleyre@lipn.univ-paris13.fr
[2] Universidade Federal de Alagoas
aydano.machado@gmail.com

Abstract. Transfer algorithms allow the use of knowledge previously learned on related tasks to speed-up learning of the current task. Recently, many complex reinforcement learning problems have been successfully solved by efficient transfer learners. However, most of these algorithms suffer from a severe flaw: they are implicitly tuned to transfer knowledge between tasks having a given degree of similarity. In other words, if the previous task is very dissimilar (respectively nearly identical) to the current task, then the transfer process might slow down the learning (respectively might be far from optimal speed up). In this paper, we address this specific issue by explicitly optimizing the transfer rate between tasks and answer to the question: "can the transfer rate be accurately optimized, and at what cost?". In this paper, we show that this optimization problem is related to the continuum bandit problem. Based on this relation, we design an generic adaptive transfer method, which we evaluate on a grid-world task.

Keywords: Reinforcement Learning, Markov Decision Processes, Transfer.

1 Introduction

In the reinforcement learning problem, an agent acts in an unknown environment, with the goal of maximizing its reward. All learning agents have to face the exploration-exploitation dilemma: whether to act so as to explore unknown areas or to act consistently with experience to maximize reward (exploit). Most research on reinforcement learning deals with this issue. Recently Strehl *et al.* [5] showed that nearly optimal strategies could be reached in as few as $\widetilde{O}(S \times A)$ time steps. However, with most real-world learning problems, the designer will face a huge state and action space, thus preventing any kind of exhaustive exploration.

One way to circumvent this problem is to use previously acquired knowledge related to the current task being learned. This knowledge may then be used to guide exploration through the state-action space, hopefully leading the agent towards areas in which high rewards can be found. This knowledge can be utilized in different ways:

- By imitation: in particular, in a multi-agent environment, agents may observe traces of other agents and use this observation to learn the environment faster [2].

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 603–611, 2012.

- By bootstrap: related tasks may have been previously learned by reinforcement [1] and the learned policy may be used to bootstrap the learning.
- By abstraction: a simplified version of the current task could have been generated to quickly learn a policy which could be used as a starting point for the current task.
- By demonstration: a human tutor may provide some explicit knowledge. Other similar settings exist in the literature, among which are “advice taking” or “apprenticeship”.

In this paper, we will focus on a simple version of the “bootstrap” transfer learning problem [1]: we will assume that a policy is available to the learner, and that this policy has been learned on a past task which shares the *same state-action space* as that of the current task. Note that unlike in the "imitation" setting, in the boostrap setting no information about the transitions in the environment is available to the learning agent.

Given this knowledge, the learning agent faces a new dilemma: it has to balance between following the ongoing learned policy and exploring the available policy. Most transfer learners do not tackle this dilemma explicitly: the amount of exploration based on the available policy does not depend on its quality. However, if the available policy is unrelated to the current task, then exploring the environment by following the available policy could result in a slowdown of the learning process. This pathological behavior has been known in the transfer learning litterature as the *negative transfer* phenomenon [7]. Ideally, this amount should be tuned such that the transfer learner be *robust* with respect to the quality of the past policy : good policies should speed up the learner while bad ones should not slow it down significantly. Recently, a new approach has been proposed to solve this issue [4]. The main idea of this approach is to estimate the similarity between the two tasks, and then to use this estimate to parameterize the transfer learning process, balancing between ongoing and past policies. However, measuring this similarity is a costly process in itself and moreover there are no guarantee that this similarity optimizes the transfer learning process.

In this paper, we show that a parameter called the *transfer rate* controlling the balance between past policy and the ongoing policy can be optimized efficiently *during* the reinforcement learning process. For this purpose, we first show in which way this optimization problem is related to the *continuum-armed bandit* problem. Based on this relation, we propose a generic adaptive transfer learning method consisting of a wrapper around some standard transfer learning algorithm, and implementing a continuum-armed bandit algorithm.

We show that under some conditions, the regret of not having chosen from the beginning the optimal value of the transfer rate can be efficiently bounded. Experiments on a grid-world task validate our approach.

The paper is organized as follows. After some preliminaries, we introduce the continuous bandit problem and relate it to the optimization of the transfer rate. The following section introduces the generic transfer learner, which is then studied in deep. Finally a set of experiments assesses both the robustness and efficiency of our approach.

2 Preliminaries

Reinforcement learning problems are typically formalized using Markov Decision Processes (MDPs). An MDP M is a tuple $\langle S, A, T, r, \gamma \rangle$ where S is the set of all states, A is the set of all actions, T is a state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$, r is a reward function $r : S \times A \rightarrow \mathbb{R}$, and $0 \leq \gamma < 1$ is a discount factor on rewards. From a state s under action a, the agent receives a stochastic reward r, which has expectation $r(s, a)$, and is transported to state s' with probability $T(s, a, s')$. A policy is a strategy for choosing actions. If it is also deterministic, a policy can be represented by a function $\pi : S \rightarrow A$. As in most transfer learning settings, we assume that the learning process is divided into episodes : at the beginning of an episode, the agent is placed on a starting state sampled from a distribution $\mathcal{D}$. The episode ends when the agent reaches a special absorbing state (the goal), or when a time limit is reached.

For any policy π, let $V^{\pi}_{M}(s)$ denote the discounted value function for π in M from state s. More formally, $V^{\pi}_{M}(s) \triangleq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$, where $r_0, r_1, \ldots$ is the reward sequence obtained by following policy π from state s. Also, let $V^{\pi}_{M} \triangleq \mathbb{E}_{s\sim\mathcal{D}}\left[V^{\pi}_{M}(s)\right]$. To evaluate the quality of an action under a given policy, the Q-value function $Q^{\pi}(s, a) \triangleq r(s, a) + \gamma \mathbb{E}_{s'\sim T(s,a,.)}\left[V^{\pi}(s')\right]$ is generally used (Here, as there are no ambiguity, M has been omitted). The optimal policy π^* is the policy maximizing the value function. The goal of any reinforcement learning algorithm is to find a policy such that the agent's performance approaches that of π^*.

To speed up learning on a new task, transfer learners exploit knowledge previously learned on a past task. Here, we will assume as in [1] that the past task and the current task have the same state-action space. We study the case where the available knowledge has the form of a policy $\bar{\pi}$ learned on the past task.

3 Transfer Learners with Static Transfer Rates

In this section, we will present a state-of-the-art transfer learner, namely PPR (*Probabilistic Policy Reuse*, as well as PPR-decay, a variation on PPR [1]). These algorithms exhibit a parameter which controls the balance between the ongoing learned policy and $\bar{\pi}$. As in many transfer methods, PPR have been directly built on a standard Q-learner, and thus share the same structure. The only difference with a Q-learner lies in the action selection method (referred here as $ChooseAction$).

Let us see how PPR works. At each step, the PPR algorithm randomly chooses to follow the policy ϵ-greedy(π) or to follow $\bar{\pi}$, as depicted in Table 1. Here, π refers to the policy induced by the Q-values ($\pi(s) = argmax_a Q_t(s, a)$) and ϵ-greedy(π) refers to the policy obtained by choosing π with probability $1 - \epsilon$, or a random action with probability ϵ. Fernandez *et al.* proposed arbitrarily to initialize φ to one at the beginning of each episode, and to decrease its influence at each step t by $0, 95^t$. PPR-decay mimics a Q-learner when $\varphi = 0$, but does not follow $\bar{\pi}$ at each step when $\varphi = 1$ because of the decay. Therefore, we introduce a variation on PPR-decay, namely PPR, in which φ is not decreased during the episode.

Table 1. Examples of $ChooseAction(s_t, \bar{\pi}, \varphi)$ functions in static transfer learners

$ChooseAction(s_t, \bar{\pi}, \varphi)$	Name of transfer algorithm
$a_t = \begin{cases} \bar{\pi}(s_t) & w.proba.\varphi \times 0,95^t \\ \epsilon\text{-greedy}(\pi) & otherwise \end{cases}$	PPR-decay [1] (PPR with exponential decay)
$a_t = \begin{cases} \bar{\pi}(s_t) & with\,proba.\,\varphi \\ \epsilon\text{-greedy}(\pi) & with\,proba\,1-\varphi \end{cases}$	PPR (Probabilistic Policy Reuse)

Clearly, φ can be seen here as a parameter controling the *transfer rate*. It is not hard to see that this rate should be dependent on the similarity between the past and the current task. Computing such a similarity is difficult in the general case, and optimizing φ can be done during learning. In this section, φ was assumed to be a constant set before the learning process. In the next sections, we will show how φ can be optimized dynamically, and ajusted after each episode.

4 Optimization of the Transfer Rate as a Stochastic Continuum-Armed Bandit Problem

Consider a transfer method such as one of those discussed above, in which a parameter $\varphi \in [0,1]$ controls the transfer rate, in such a way that if $\varphi = 0$, the policy $\bar{\pi}$ is not being used, and if $\varphi = 1$, the agent follows exclusively $\bar{\pi}$. Let us consider the problem of optimizing φ, in order to improve the speed up learning. For the sake of simplicity, adjustment of φ will occur only after each episode, thus exploiting the sequence of rewards gathered during the last episode.

Consider a learning episode starting at time t. Before the episode begins, the agent has to choose a value of φ, which ideally would yield the highest expected gain $V_t(\varphi) \triangleq \mathbb{E}\left[r_t + \gamma r_{t+1} + \ldots \mid \varphi\right]$. At the end of the episode, the agent can compute $\sum_k r_{t+k}\gamma^k$ which is an unbiased estimator of $V_t(\varphi)$. Choosing the best value for φ is challenging, as gradient methods which require the knowledge of $\frac{\partial V_t}{\partial \varphi}$ might not be applicable. It turns out that this problem is a typical *continuum armed bandit problem.*

The *continuum armed bandit* problem which belongs to the well known family of multi-armed bandit problems, is a particularly appropriate setting for the optimization of $V_t(\varphi)$. In this setting, at each time step t, a learner chooses a real number $X_t \in [0,1]$ and receives a reward depending on the sequence $X_1 \ldots X_t$. The goal of the learner is to maximize the total sum of rewards, or to minimize the regret as stated formally below:

Definition 1. ***(The continuum armed-bandit problem)*** *Let $P(. \mid x, t)$ be an unknown distribution indexed by $x \in [0,1]$ and $t \in \{1 \ldots n\}$. At each trial t, the learner chooses $X_t \in [0,1]$ and receives return Y_t randomly drawn from $P(. \mid X_t, t)$. Let $b_t(x) = \mathbb{E}\left[Y_t \mid X_t = x, t\right]$. The agent's goal is to minimize its expected regret $\mathbb{E}\left[\sum_t b_t(x^*) - \sum_t Y_t\right]$, given that $x^* = \sup_{x\in[0,1]} \sum_{t=1}^{n} b_t(x)$.*

This definition is a slight generalization of that found in [3]. Still in [3], Kleinberg designs an algorithm called *CAB1* solving this continuum armed bandit problem with the following guaratees:

Corollary 2. *If the function b_t is L-lipschitz (i.e., $\mid b_t(x) - b_t(x') \mid \leq L \mid x - x' \mid$ for all $x, x' \in [0,1]$), then using CAB1 yields an expected regret bounded by $O(Ln^{\frac{2}{3}} \log^{\frac{2}{3}} n)$.*

In the next section, we will describe AdaTran, an algorithm using CAB1 as a subroutine. Thus, corollary 2 will later be useful to derive a regret bound on AdaTran.

5 AdaTran: A Generic Adaptive Transfer Framework

We now present a generic adaptive transfer learning algorithm, which can be seen as a wrapper around a transfer learner, optimizing the transfer rate φ using a *stochastic adversarial continuum armed-bandit* algorithm refered to as *UpdateContBandit*. This leads to the *AdaTran* wrapper, a generic adaptive transfer algorithm in which many transfer learners can be implemented. Note that even though most transfer learners do not have such a parameter, they can often be modified so as to make φ appear explicitly.

Algorithm 1. AdaTran

```
Init()
$t \leftarrow 0$
$\varphi \leftarrow \varphi_0$
for each episode $h$ do
    set the initial state $s$
    while (end of episode not reached) do
        $a_t = ChooseAction(s_t, \bar{\pi}, \varphi)$
        Take action $a_t$, observe $r_{t+1}, s_{t+1}$
        $Learn(s_t, a_t, r_{t+1}, s_{t+1})$
        $t \leftarrow t + 1$
    end while
    $\varphi \leftarrow UpdateContBandit(\bar{\pi}, \varphi, \langle r_1, r_2, \ldots \rangle)$
end for
```

Depending of the function used for *ChooseAction* (e.g. one of Table 1), *Learn* (e.g. a TD update of a model-based learning step) and *UpdateContBandit* (e.g. CAB1), the AdaTran will lead to different types of transfer learners. In particular, the experimental section will evaluate AdaTran(PPR) and AdaTran(PPR-decay) Let us now show how the bound on the regret of CAB1 can be applied. Let t_i be the time at which the i^{th} episode begins. Suppose $V_t(\varphi)$ satisfies the L-lipschitz condition. Let φ_t refer to the parameter chosen by CAB1 at time t. Then on the n first episodes, we have $\sum_{i=1}^{n} V_{t_i}(\varphi^*) - V_{t_I}(\varphi_t) \leq O(Ln^{\frac{2}{3}} \log^{\frac{2}{3}} n)$ iff the following assumption holds:

Assumption 3. *At any given time step t, the value functions $V_t(\varphi)$ does not depend on previous actions in the MDP.*

Equivalently, we might assume that the sequence of functions $V_t(\varphi)$ is fixed in advance. Note that this type of assumption has been widely discussed in the multi-armed bandit setting. Also, there has been some attempts to overcome this assumption in the bandit literature, in particular [6]. These attempts usually rely on non standard definition of the regret and/or on strong assumptions on the type of non-stationarity of the environment, which does not suit our setting. Nevertheless, in our case, the assumption 3 seems reasonable since in most situations, choosing a sub-optimal exploration strategy for a given episode will not jeopardize the whole learning process.

We have seen in this section that optimizing φ with bounded regret may be possible, given that $V_t(\varphi)$ satisfies the Lipschitz condition. This remains to be proven. We will show this in detail for AdaTran(PPR).

6 Properties of the Value Function

In order to bound the regret of AdaTran, we now need to study the properties of the value function $V(\varphi)$ (the parameter t will be omitted). We will conduct this analysis in detail for AdaTran(PPR).

First, we will show that without any restrictions, $V(\varphi)$ cannot be optimized in the worst case. This is due to the fact that the function $V(\varphi)$ can be made arbitrarily close to any continuous function. To show this, we must first recall what Bernstein polynomials are. Without loss of generality, we will assume that the probability distribution $\mathcal{D}$ of starting states is equal to one on a given state s_0 and is null elsewhere.

Definition 4. *For any function f on $[0,1]$, the associated Bernstein polynomial is defined as follows: $B_n(f,x) \triangleq \sum_{k=0}^{n} f(\frac{k}{n}) b_{k,n}(x)$, where $b_{k,n}(x) \triangleq \binom{n}{k} x^k (1-x)^{n-k}$.*

Unfortunately, the set of all possible function $V(\varphi)$ includes the set of all Bernstein polynomials:

Lemma 5. *Let f be any continuous function of $[0,1]$ and $d \in \mathbb{N}$. Then there exist an MDP such that $V(\varphi) = B_d(f,\varphi)$*

Proof. Let A^d be a deterministic MDP having the structure of a binary tree of depth d. Let the root state be s_0. At each state (node in the tree), A^d allows two actions *left* and *right* leading respectively to the left and right child states. Let the rewards of all state-actions be null, except those between depth $d-1$ and d. Let us allocate rewards to the 2^d states-actions pairs at depth $d-1$ of the tree in the following way: if a state-action pair can be reached from the root by l steps left and $d-l$ steps right (in any order), then its reward must be $\gamma^{1-d} \times f(\frac{l}{d})$. Suppose on each state s, the standard policy is $\pi(s) = right$, whereas the transfer policy is $\bar{\pi}(s) = left$. Thus, a learner exploring this tree and starting from the root node would walk down the tree, choosing randomly left and right branches with probability φ (respectively $1-\varphi$), and collecting a reward $\gamma^{1-d} \times f(\frac{i}{d})$ at the bottom of the tree. Let $V^{A^d}(\varphi) = \mathbb{E}\left[r_1 + \gamma r_2 + \ldots\right]$ be the value of the root node, parameterized by φ. Clearly, the probability that an agent chooses l times the *left* action and $d-l$ times the *right* action in a given order is $\varphi^l(1-\varphi)^{d-l}$. Thus, the probability of choosing l times *left* in any order is $b_{l,d}(\varphi) = \binom{d}{l}\varphi^l(1-\varphi)^{d-l}$. Therefore, we have $V^{A^d}(\varphi) = \frac{1}{\gamma^{d-1}}\sum_{l=0}^{d}\gamma^{1-d} \times f(\frac{l}{d}) \times b_{l,d}(\varphi)$.

Recall now that the Weierstrass theorem states that for any continuous function f on $[0,1]$, $B_n(f,x)$ converges uniformly to $f(x)$ as $n \to \infty$. An immediate corollary is:

Corollary 6. *For any continuous function f on $[0,1]$, for any $\epsilon > 0$, there exists a deterministic MDP such that $|V(\varphi) - f(\varphi)| < \epsilon$ for all φ.*

This has important implications in our setting: this corollary tells us that $V(\varphi)$ can be arbitrarily close to any continuous function, provided the appropriate MDP. Thus, optimizing $V(\varphi)$ without further restrictions is hopeless.

However, by upper-bounding the rewards, we will now show that $V(\varphi)$ finally satisfies the lipschitz condition. Let us first bound the derivative of Bernstein polynomials.

Lemma 7. *Let f be a real-valued function on $\{0, \frac{1}{n}, \frac{2}{n}, \ldots, 1\}$. Then we have $\sup_{x\in[0,1]} \mid \frac{d}{dx}B_n(f,x) \mid \leq 2n \sup \mid f(x) \mid$.*

Due to space constraints, the proof of this technical lemma is omitted.

Applying this lemma on the tree MDPs used in lemma 5, we get the bound $\mid \frac{d}{dx}V^{A_d}(\varphi) \mid \leq 2d\gamma^{d-1}r_{max}$, given that all rewards are bounded by r_{max}. Finally, we generalize this result (again, the proof is omitted).

Proposition 8 *For any MDP M in which rewards are bounded by r_{max} , any policies π and $\bar{\pi}$, and a starting state s_0, we have $\left|\frac{d}{d\varphi}V(\varphi)\right| \leq \frac{2r_{max}}{(1-\gamma)^2}$.*

Finally, combining the above result with the regret bound of corollary 2, we can now show the following:

Corollary 9. *The cumulative regret per episode of AdaTran(PPR) is $O\left(\frac{r_{max}h^{\frac{2}{3}}\log^{\frac{2}{3}}h}{(1-\gamma)^2}\right)$, where h is the episode number.*

Again, note that the regret does not depend on the number of states, so the MDP might be huge here.

7 Experiments

In this section, we evaluate AdaTran on a standard benchmark for transfer learning: the grid-world problem [1,2]. The reason behind our choice of this learning task lies in its simplicity. In this learning task, an agent moves in a 25×25 two-dimensional maze. Each cell of this grid-world is a state and it may be surrounded by zero to four walls. At each time step, the agent can choose to move from its current position to one of the reachable contiguous north/east/west/south cell. If a wall lies in between, the action fails. Otherwise, the move succeeds with probability 90%, and with probability 10%, the agent is randomly placed on one of the reachable cells contiguous to the current cell. At the beginning of each episode, the agent is randomly and uniformly placed on the maze. As the agent reaches the goal state (the exit of the maze), it is given a reward of 1, and the episode is ended. All other rewards are null and the discount factor is arbitrarily set to $\gamma = 0,95$.

The goal of the current task (the exit) is to reach the bottom right corner. We generated two other tasks based on the exact same maze but different goals. The first task refered to as the "similar task" has its goal located two cells away from the bottom right corner, whereas the second task, refered to as the "dissimilar task", has its goal located on the opposite corner (upper left corner).

The optimal policies computed on each of these two tasks will serve as transfer knowledge to solve the current task. The goals of the "similar task" and the current task are very close to each other. Thus, transfer between both might be highly valuable. On the opposite, the goals of the "dissimilar task" and that of the current task are very dissimilar to one another, and transfer is likely to be less valuable.

AdaTran is compared to other algorithms on figure 1 and 2. Each of these curves have been averaged over 100 runs. The x-axis represents the episodes, and the y-axis is the average episode length, given that episode are limited to 10000 steps.

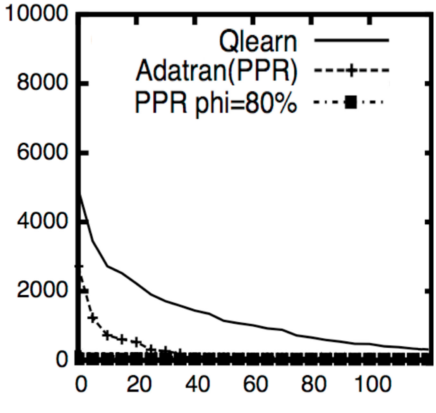


Fig. 1. Similar transfer tasks

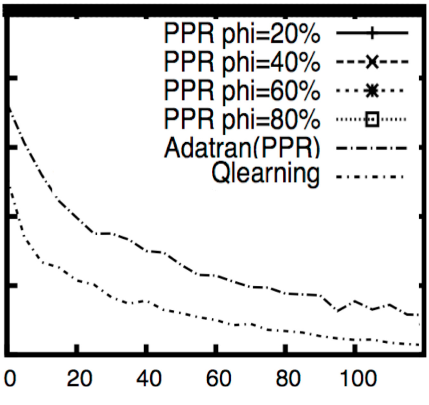


Fig. 2. Dissimilar transfer tasks

When transferring from the "similar task", PPR performs extremely well, and AdaTran performs much better than Q-learning, as it quickly finds that tasks are similar, but a bit worse that PPR as expected. When transferring from the "dissimilar task", PPR with various levels of φ perform the worse: all episodes reach 10000 steps in average, and the exit of the maze is nearly never found. The best learner here is Q-learning, which ignores the transfer policy. In between both lies AdaTran, which quickly detects that the transfer policy should not be trusted.

Clearly, AdaTran is shown to be robust to dissimilar tasks unlike most other transfer methods, and it is shown to transfer successfully a high amount of knowledge on similar tasks.

8 Conclusion

In this paper, we have presented a new framework for explicitly optimizing the transfer rate in reinforcement learning. We have shown how this framework could be applied on a well known transfer learner to make the transfer rate auto-adaptable, namely the *PPR* method. Moreover, by bounding the maximum reward, we showed that the average regret converged towards zero.

References

1. Fernandez, F., Veloso, M.: Probabilistic Policy Reuse in a Reinforcement Learning Agent. In: The Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS (2006)
2. Price, B., Boutilier, C.: Accelerating Reinforcement Learning through Implicit Imitation. Journal of Artificial Intelligence Research 19, 569–629 (2003)
3. Kleinberg, R.: Nearly tight bounds for the continuum-armed bandit problem. In: Advances in Neural Information Processing Systems, vol. 17, pp. 697–704 (2004)
4. James, L., Carroll, K.S.: Task Similarity Measures for Transfer in Reinforcement Learning Task Libraries. In: The 2005 Int. Joint Conference on Neural Networks (2005)
5. Strehl, A.L., Li, L., Wiewiora, E., Langford, J., Littman, M.L.: PAC model-free reinforcement learning. In: ICML, vol. 06, pp. 881–888 (2006)
6. Pucci de Farias, D., Megiddo, N.: Combining expert advice in reactive environments. Journal of the ACM 53(5), 762–799 (2006)
7. Rosenstein, M.T., Marx, Z., Kaelbling, L.P., Dietterich, T.G.: To transfer or not to transfer. In: NIPS 2005 Workshop on Transfer Learning, Whistler, BC (2005)
8. Lazaric, A., Restelli, M., Bonarini, A.: Transfer of Samples in Batch Reinforcement Learning. In: Proceedings of the Twenty-Fifth International Conference on Machine Learning, Helsinki, Finland, pp. 5–9 (2008)

Adaptive Classifier Selection in Large-Scale Hierarchical Classification

Ioannis Partalas*, Rohit Babbar, Eric Gaussier, and Cecile Amblard

LIG, Université Joseph Fourier, Grenoble 1
Grenoble, cedex 9, 38041, France
firstname.lastname@imag.fr

Abstract. Going beyond the traditional text classification, involving a few tens of classes, there has been a surge of interest in automatic document categorization in large taxonomies where the number of classes range from hundreds of thousands to millions. Due to the complex nature of the learning problem posed in such scenarios, one needs to adapt the conventional classification schemes to suit this domain. This paper presents a novel approach for classifier selection in large hierarchies, which is based on exploiting training data heterogeneity across the hierarchy. We also present a meta-learning framework for further flexibility in classifier selection. The experimental results demonstrate the applicability of our approach, which achieves accuracy comparable to the state-of-the-art and is also significantly faster for prediction.

Keywords: Hierarchical Classification, Classifier Selection, Meta-learning.

1 Introduction

Many recent practical applications of text classification have the number of target classes as an added dimension to the complexity of the underlying learning problem. Directory Mozilla and Wikipedia exemplify this research challenge in the domain of text classification, where the number of target classes range from hundreds of thousands to over a million. Due to the enormous effort involved in manually classifying unseen data in such scenarios, automatic classification has assumed significant importance. For large scale classification, an underlying semantic structure, a rooted tree for instance, typically exists among the classes. The taxonomy structure serves as useful prior information aimed at improving classification accuracy and speed. If no semantic structure exists among the classes or is ignored if it exists, one needs to evaluate $O(K)$ one-vs-all classifiers, one for each of the K classes. This technique, also referred to as flat classification, consequently leads to a significant slowdown in prediction performance.

In hierarchical classification, major challenges in obtaining higher classification accuracy include: (i) error propagation, since the overall classification mechanism cannot recover from classification error at top levels of the hierarchy, (ii)

* I. Partalas and R. Babbar equally contributed to this work.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 612–619, 2012.

data heterogeneity across levels in the hierarchy, and (iii) class size imbalance between positive and negative examples for one-vs-all classification. Another important aspect for large scale hierarchical classification, which is also the main focus of this work, is prediction speed. This factor has largely been ignored while designing classification mechanisms in this domain, but is crucial for acceptable behavior in many applications, such as large scale Question-Answering systems.

2 Related Work and Our Contributions

Various approaches have been proposed for hierarchical classification, which can be broadly divided into one of the two techniques: (i) Big-bang approaches, which train a single classifier for the entire hierarchy and hence are more suited for relatively small-scale problems as in [3] where the number of classes are limited to 1172, and (ii) Top-down approaches, in which the test document is classified at the root and then iteratively following the most confident child class, till leaf node is reached. In [5], an SVM classifier is deployed at each node of the hierarchy, which is relatively slow for training and testing. Deep classification technique [9] is slightly better in terms of accuracy but suffers from the limitation that it needs to train a new classifier for every test document. Refined Experts [2] employs top-down SVM classifiers which are augmented with a bottom-up pass using validation set to correct false negatives. Classifier selection for hierarchical classification on a smaller scale in protein function prediction has been studied in [8]. A related study, with focus on empirical trade-offs of large scale hierarchical classification, has been explored in [1].

This work presents a classification technique which exploits the properties of the data, such as feature to example ratio and data imbalance between the target class and rest of the classes, in a large scale hierarchy. Based on such properties, our algorithm performs automatic classifier selection by choosing either an SVM or a Naive Bayes classifier at the classification nodes of the hierarchy. Additionally, this work formulates and solves a meta-learning problem for dynamic classifier selection. Section 4 presents in detail the criteria which determine the choice of classifiers.

3 Problem Setup

In single-label multi-class hierarchical classification, the training set can be represented by $S = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$. In the context of text classification, $\mathbf{x}^{(i)} \in \mathcal{X}$ denotes the vector representation of document i in the input space $\mathcal{X} \subseteq \mathbb{R}^d$. Assuming that there are K classes denoted by the set $\mathcal{Y} = \{1 \ldots K\}$, the label $y^{(i)} \in \mathcal{Y}$ represents the class associated with the instance $\mathbf{x}^{(i)}$. The hierarchy in the form of rooted tree is given by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} \supseteq \mathcal{Y}$ denotes the set of nodes of $\mathcal{G}$, and $\mathcal{E}$ denotes the set of edges with parent-to-child orientation. The leaves of the tree which usually forms the set of target classes is given by $\mathcal{Y} = \{u \in \mathcal{V} : \nexists v \in \mathcal{V}, (u, v) \in \mathcal{E}\}$.

In the above setup, given a new test instance $\mathbf{x}$, the goal is to predict the class $\widehat{y}$. This is typically done by making a sequence of predictions iteratively in a top-down fashion starting from the root until a leaf node is reached. At each non-leaf node $v \in \mathcal{V}$, a score $f_c(\mathbf{x}) \in \mathbb{R}$ is computed for each child c and the child $\widehat{c}$ with the maximum score is predicted i.e. $\widehat{c} = \underset{c:(v,c)\in\mathcal{E}}{\text{argmax}} f_c(\mathbf{x})$.

In addition to requiring *accurate* predictions, we also focus on *prediction speed*, two seemingly contradicting design requirements for a machine learning algorithm. To address the above conflicting requirements of high prediction accuracy and faster prediction and training time, we focus on Support Vector Machine (SVM) and Naive Bayes (NB) classifiers as base classifiers. SVM classifier is known to have better accuracy but is slower to train and deploy for prediction. NB classifier, on the other hand, is faster for training and prediction but is on the lower side of the accuracy spectrum. In the next section, we present conditions in large hierarchies, which determine the selection of classifiers to achieve better run-time performance without sacrificing accuracy.

4 Classifier Selection in LSHC

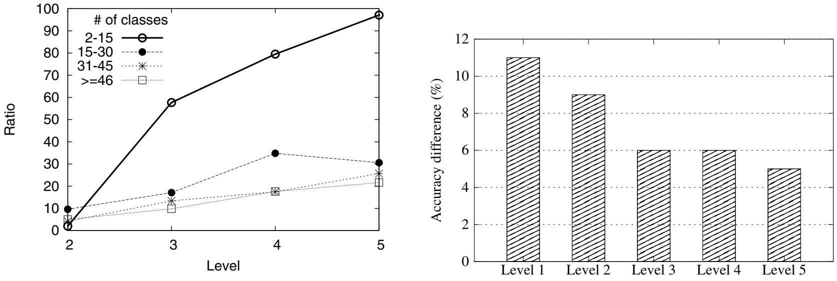
In this section, we present two approaches to classifier selection that exploit the data heterogeneity in large scale hierarchical classification. The first one, referred to as static approach [1], is based on using the relation between number of features and number of training examples for the classification problem at a given node in the hierarchy. The second approach to classifier selection is based on solving a meta-learning problem which includes further meta-features such as test instance size and class imbalance in one-vs-all classification.

4.1 Static Approach to Classifier Selection

For a multi-class classification problem at node v of the hierarchy, let d_v denote the dimensionality of the feature space and n_v denote the number of training documents for which the root-to-leaf path goes through node v. Let their ratio for node v in the hierarchy be denoted by r_v, i.e. $r_v = \frac{d_v}{n_v}$.

As one traverses down from the root node towards the leaves, in large scale hierarchies such as DMOZ, r_v varies over a wide range of values. Due to large number of classes corresponding to the leaf nodes, the number of training documents (n_v) decrease much faster than the number of features (d_v) for classification nodes on root to leaf path. As a result, this ratio is much higher at hierarchy levels close to the root as compared to its value for nodes at lower levels. Figure 1(a) shows the variation of average value of r_v for DMOZ dataset when plotted against the hierarchy levels. Each piece-wise linear curve in the plot corresponds to the class size range of the multi-class problem. Two important properties of the dataset, one of which follows from Figure 1(a), are:

[1] Called Adaptive Classifier Selection in our previous work [1].



(a) Ratio variation

(b) Accuracy differences

Fig. 1. 1(a)Variation in ratio of feature set size to training sample size with the hierarchy level, and 1(b) Difference of SVM and NB accuracy, (SVM - NB), in % for each hierarchy level. Level 1 corresponds to the root and level 5 to level leading to leaves.

1. The ratio r_v increases towards the leaves;
2. Almost 97% of the multi-class classification problems involve 2-15 classes.

The above observations imply that in order to achieve the goal of high accuracy and faster run-time, one needs to exploit the wide variation in training data properties. In this context, we next present the relevant results from statistical learning theory which deal with accuracy measure of discriminative classifiers (such as SVM) and generative classifiers (such as NB).

Let f_G and f_D represent the classifiers learned by fitting generative and discriminative model respectively and $f_{G,\infty}$ and $f_{D,\infty}$ be their corresponding asymptotic versions, i.e. functions learned when the sample size approaches infinity. Let $\varepsilon(.)$ be the function representing the generalization error of its argument. For a classification problem in d-dimensional feature space with n training examples, these results can be summarized as follows [6]:

1. $\varepsilon(f_{D,\infty}) \leq \varepsilon(f_{G,\infty})$;
2. $\varepsilon(f_G) \leq \varepsilon(f_{G,\infty}) + \delta_0$ if $n = \Omega(\ln(d))$;
3. $\varepsilon(f_D) \leq \varepsilon(f_{D,\infty}) + \delta_0'$ if $n = \Omega(d)$;
   for arbitrary but fixed $\delta, \delta_0' > 0$; $\Omega(.)$ denotes the big Omega notation.

Informally, the above inequalities can be interpreted as follows:

1. The generalization performance of discriminative classifiers is better than that of generative classifiers under asymptotic regime of operation.
2. The number of training examples required for discriminative classifier to reach its asymptotic performance is at least *linear* in the number of features.
3. The number of training examples required for generative classifier to reach its asymptotic performance is at least *logarithmic* in the number of features.

The above results show that even though SVM is a better classifier for nodes close to the root, NB can be used for nodes in the lower levels of the hierarchy due to its faster training and prediction time. Figure 1(b) confirms this intuition further, showing that NB accuracy is much closer to that of SVM for lower levels (4 and 5) than at higher levels (1 and 2). Consistently lower accuracy of NB for all levels in the hierarchy can be attributed to the argument indexed 1 above.

In order to allow further flexibility in classifier selection and instead of fixing NB classifier for the *entire* lower levels, we can use a threshold value τ_v for ratio r_v to choose the classifier at node v, such that

$$\text{Classifier at node } v = \begin{cases} \text{Naive Bayes} & \text{if } r_v \geq \tau_v \\ \text{SVM} & \text{otherwise} \end{cases}$$

The thresholding strategy, even though a simplification of the three arguments presented above, works well in practice, as shown in the experimental section 6.

4.2 Adaptive Hierarchical Classifier Selection

Using the classifier selection strategy introduced in the previous section, the choice of a classifier at every node in the hierarchy becomes fixed for all test instances. Further adaptivity can be achieved by enabling classifier selection for each test instance separately. The rationale of incorporating an adaptive selection mechanism is that classification models have different levels of expertise in different parts of the feature space. For example, for a specific node in the hierarchy, a NB classifier may be complementary to an SVM classifier. Thus, it would be desirable to select NB in some test cases targeting to both predictive accuracy and computational performance. The proposed method for hierarchical classifier selection is based on the concept of meta-learning. One of the objectives of meta-learning methods is to produce general and robust learning algorithms [7].

In the context of hierarchical classification, the purpose of the meta-learning framework is to select the best classifier for each node in the hierarchy as we traverse it in a top-down manner. In order to do so, we need to define an appropriate meta-learning problem. Let $S_V = \{(\mathbf{x}_V^{(i)}, y_V^{(i)})\}_{i=1}^{N_V}$ be a validation set containing examples of the initial classification problem and C_a, C_b be two classifiers that have been trained for each multi-class problem in the hierarchy. For every example in the validation set, we create meta-learning examples by traversing the hierarchy top-down. For a specific node v and a validation example i, a meta-learning example is defined by the tuple: $\mathbf{x}_m^{(i,v)} =< \gamma_v, r_v, ubr_v, sz_i >$ such that the elements of the tuple are the meta-features of the meta-learning problem:

- γ_v : number of children of node v.
- r_v : ratio of the number of features to number of training examples for node v.
- ubr_v : the unbalanced ratio of the classification problem at node v, calculated as $\frac{\#\text{examples of minority class}}{\#\text{examples of majority class}}$
- sz_i : the size of instance i. Different classification algorithms behave differently according to the size of the instance and thus it would be desirable to adapt to this phenomenon.

Next, we need to define the label space of the meta-learning problem. Different learning problems can be defined according to the characteristics of the classifiers (prediction accuracy, training and testing time) and the objectives of the hierarchical classification problem (accuracy, computational cost). For example, if we assume that $C_a \prec C_b$ in terms of prediction accuracy and $C_a \succ\succ C_b$ in terms of computational cost, where $\prec$ denotes the natural preference relation, then we set the following learning problem:

$$y_m^{(i,v)} = \begin{cases} C_a & \text{if } C_a \text{ is correct} \\ C_b & \text{otherwise} \end{cases}$$

In this problem the objective is to identify regions where C_a complements C_b while reducing the computational cost in cases where both classifiers have good or bad prediction accuracy. Note that the definition of the meta-learning problem is flexible in order to allow different instantiations by considering different and multiple classification schemes. For example, one may consider multiple classifiers and form a multi-label problem where more than one classifier could be correct for an instance. In this case the method could be coupled with ensemble techniques for acquiring a decision.

5 Experimental Setup

For the experiments we use the publicly available DMOZ data set from LSHTC2 [2]. The dataset, after preprocessing by stemming and stopword removal, appears in the LibSVM format. Table 1 presents the important properties of the dataset. Since the average number of labels per document is 1.02, we consider it as single-label classification for our purpose. We use Liblinear [4] to train the models for L2-regularized L2-loss support vector classification. The models are trained for all 7,574 non-leaf nodes for One-Vs-All classification. For NB classifier, we implement the standard multinomial NB using Laplace smoothing.

Table 1. Training Data Properties

Property Name	**Value**
Total number of training examples	394,756
Size of the Overall Feature Space	594,158
Number of Target Classes ($|\mathcal{Y}|$)	27,875
Number of Nodes in the Hierarchy ($|\mathcal{V}|$)	35,449
Total number of multi-class classifiers	7,574

For producing the meta-learning training set we use a validation set (separate from the training set for building the classifiers) containing 3000 examples. As

[2] http://lshtc.iit.demokritos.gr

meta-learner, we use a decision tree based on C4.5 algorithm as it can provide interpretable rules. The Weka machine learning library was used in this case setting the confidence factor to 0.25, with reduced error pruning and without Laplace smoothing, using 10-fold cross-validation.

6 Results and Analysis

Table 2 shows the different classification mechanisms and the metrics of interest for the overall classifier, which include, (i) SVM classifier for the entire hierarchy (SVM-TD), (ii) Static classifier selection strategy based on threshold value (SCS-τ), (iii) adaptive hierarchical classifier selection method (AH-CS) for C_a =NB and C_b=SVM, and (iv) NB classifier for the entire hierarchy (NB-TD). We first notice that the best performing algorithm in terms of accuracy is AH-CS with a slight difference over SVM-TD, while the gain in prediction speed is about 5 times. Note that in the case of AH-CS the training time of the models is the sum of the training times of SVM-TD and NB-TD as we need to retain all the models due to the instance-based nature of the method.

Table 2. Trade-off between Prediction Accuracy in %, Total Training for entire dataset in hours, and Average Test Time per Instance in seconds

Method	**Accuracy (%)**	**Tr. Time (hours)**	**Test Time (secs)**
SVM-TD	35.58	35	20
SCS-τ, $\tau = 60$	35.19	22	12
SCS-τ, $\tau = 30$	34.68	12	5
AH-CS	**35.66**	35.25	**4**
NB-TD	22.22	0.25	0.5

AH-CS selected 45.58% times NB models during the testing procedure. To better understand the results of the proposed method we consider the contingency matrix of the two classifiers for each level of the hierarchy. Noticeably, the NB classifier corrects the errors of SVM in 6.8%, 6.4%, 6.6% and 6.7% of the examples for the first 4 levels respectively. This shows that being able to identify these cases can lead us to better performance both in accuracy and speed. We also calculated the κ-statistic for each level which shows how diverse the classifiers are, getting 0.42, 0.45, 0.44 and 0.45 for the 4 levels respectively. $\kappa = 1$ means that the two classifiers are identical while $\kappa = 0$ indicates independent classifiers. The results show that the classifiers are diverse across the hierarchy and supports the rationale of using different classifiers for different cases. Note that the achieved performance of our method is comparable to the best participant (38.8%) in LSHTC for the DMOZ track. However, the objective of our work does not coincide with the participants' in the challenge since their major focus is on accuracy related metrics. As a result, some of them may employ some pre-processing steps to boost accuracy.

From rows 1 and 3 of table 2, the accuracy of the hierarchical classifier by using static classifier selection is comparable to top-down SVM, while being four times faster in prediction and three times faster to train. SCS-τ was computed based on a uniform threshold value of $\tau_v = 60$ and $\tau_v = 30$, $\forall v \in \mathcal{V}$. Increasing the threshold value selects more SVM classifiers, leading to better accuracy but slower training and test time, while decreasing it would have the opposite effect.

The gain in speed-up for test time is achieved as a result of more compact models built by NB as compared to SVM from same the training data. All NB models can be loaded in the physical memory for predictions. For SVM, the models for only the top two levels can be loaded in physical memory.

7 Conclusions and Future Work

We presented a static and an adaptive classifier combination technique to build large scale hierarchical classification systems. As a result, we not only achieve accuracy comparable to state-of-the-art but also reduce the prediction time significantly compared to the top-down SVM classifier. Further work includes the consideration of more complex topologies such as directed acyclic graphs and also allowing multiple labels.

Acknowledgement. This work was supported by the French National Research Agency (ANR) as part of the project Class-Y (ANR-10-BLAN-02).

References

1. Babbar, R., Partalas, I., Gaussier, E., Amblard, C.: On empirical tradeoffs in large scale hierarchical classification. In: ACM CIKM (2012)
2. Bennett, N.P., Nguyen, N.: Refined experts: improving classification in large taxonomies. In: Int. ACM SIGIR Conference, pp. 11–18 (2009)
3. Cai, L., Hofmann, T.: Hierarchical document categorization with support vector machines. In: CIKM, pp. 78–87 (2004)
4. Fan, E.R., Chang, W.K., Hsieh, J.C., Wang, R.X., Lin, J.C.P.: LIBLINEAR: A library for large linear classification. JMLR 9, 1871–1874 (2008)
5. Liu, Y.T., Yang, Y., Wan, H., Zeng, J.H., Chen, Z., Ma, Y.W.: Support vector machines classification with a very large-scale taxonomy. SIGKDD Explor. Newsl., 36–43 (2005)
6. Ng, Y.A., Jordan, I.M.: On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In: NIPS, pp. 841–848 (2001)
7. Schaul, T., Schmidhuber, J.: Metalearning. Scholarpedia 5, 4650 (2010)
8. Secker, A., Davies, N.M., Freitas, A.A., Clark, B.E., Timmis, J., Flower, R.D.: Hierarchical classification of g-protein-coupled receptors with data-driven selection of attributes and classifiers. Int. J. Data Min. Bioinformatics, 91–210 (2010)
9. Xue, R.G., Xing, D., Yang, Q., Yu, Y.: Deep classification in large-scale text hierarchies. In: Int. ACM SIGIR Conference, pp. 619–626 (2008)

An Efficient Algorithm for Anomaly Detection in a Flight System Using Dynamic Bayesian Networks

Mohamad Saada and Qinggang Meng

Department of Computer Science
Loughborough University
Loughborough
United Kingdom
{M.Saada,Q.Meng}@lboro.ac.uk

Abstract. Despite the fact that Dynamic Bayesian Network models have become a popular modelling platform to many researchers in recent years, not many have ventured into the realms of data anomaly and its implications on DBN models. An abnormal change in the value of a hidden state of a DBN will cause a ripple-like effect on all descendent states in current and consecutive slices. Such a change could affect the outcomes expected of such models. In this paper we propose a method that will detect anomalous data of past states using a trained network and data of the current network slice. We will build a model of pilot actions during a flight, this model is trained using simulator data of similar flights. Then our algorithm is implemented to detect pilot errors in the past given only current actions and instruments data.

Keywords: Anomaly Detection, Dynamic Bayesian Networks, Intelligent Systems, Machine Learning.

1 Introduction

Data anomalies can occur due to many different reasons, it can be due to a sensor reading error, or a communication error while data is being transmitted through a network, it can be a new type of a network attack which has not been encountered before, or it can simply be due to unexpected and rare system behaviour. The list goes on but in all cases data anomalies share the property of being a type of data that is deviated away from normal data patterns. Anomaly Detection is a very helpful instrument wherever it is applied, banks detect suspicious invalid transactions through the use of anomaly detection, it helps maintain a good level of Quality of Service in network communications, or detect a failing components of a mechanical system. Anomaly detection is useful in endless scenarios, and that is why it has been implemented and investigated in many ways, one of which is through the use of probabilistic graphical models. Many researchers have taken different approaches to solve the anomaly detection problem, but most of these approaches belong to three main streams used to solve the problem of data anomaly detection as shown by [1,2], we list these below:

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 620–628, 2012.

- **Unsupervised Approach:** Works by detecting anomalies without having any prior information about them.
- **Supervised Approach:** Works by modelling both anomalous data and normal data, through manually specifying which data is considered anomalous and which is normal.
- **Semi-supervised Approach:** Works by modelling normal data only, and then uses the modelled data in the algorithm that detects anomalous data.

Not many researchers have Dynamic Bayesian Network models as basis for their approach, in [3] researchers have developed two machine learning methods a coupled and uncoupled DBN Anomaly Detector which aim to detect erroneous data in two different windspeeds data streams in real time. These methods can work on single or multiple data streams in real time. And in [4] researchers suggested an anomaly detection algorithm based on the use of a new implementation of the Dirichlet process precision parameter, outlier detection is done by calculating a maximum a posteriori (MAP) of the data partition, where observations forming small or singleton clusters are deemed as anomalies. Researchers in [5] have used a Bayesian Network to model the outliers as an "unlikely events under the current favored theory of the domain", their approach is based on using a Bayesian network modelling the background knowledge coupled with two rules to detect the outliers, their approach not only focuses on detecting outliers but also on explaining why these data are considered outliers. Researchers in [6] use an unsupervised approach towards detecting fraud operations in a stock exchange market, they use Peer Group Analysis (PGA) technique which is concerned with characterizing the expected pattern of behaviour around the targeted time series financial sequence in terms of the behaviour of similar objects and then detect outliers through the detection of difference in evolution of the actual behaviour and expected behaviour.

When using probabilistic models such as Dynamic Bayesian Networks to model a certain environment or a system, modellers usually work with large amounts of data, and that is due to the fact that the main reason behind probabilistic modelling is to extract useful information about the environment from its data, which otherwise cannot be easily extracted or interpreted, not even by experts. And as we know wherever there is manipulation of large amounts of data there are bound to be data anomalies which can have damaging effects on the processes manipulating this data which could lead to erroneous outcome. And these data anomalies can affect the information that is gathered from the model greatly, and this is why in this paper we are focusing on the detection of data anomalies that occur while working with Dynamic Bayesian Networks which will be denoted from here on as DBN.

2 Dynamic Bayesian Network Model

A Dynamic Bayesian Network is the extension of Bayesian Networks to model probability distributions of sets of random variables over time [7]. Bayesian Network on the other hand are a type of probabilistic models that are based on

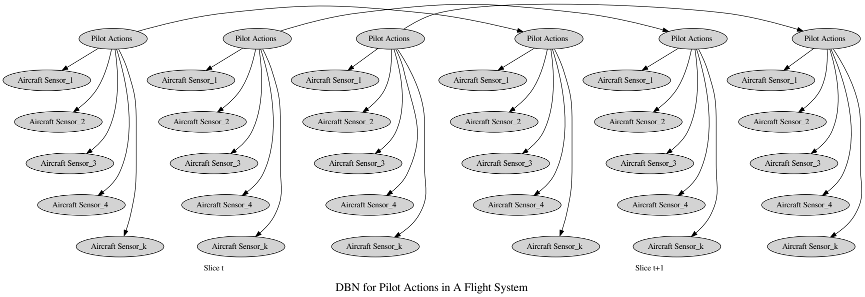


Fig. 1. A Simplified DBN Model for Pilot Actions in a Flight System

directed acyclic graphs (DAGs) [8], the nodes in this model represent propositional variables of interest and the links between them represent the dependencies among these variables [8], and these dependencies are quantified by conditional probabilities of each node given its parents in the network. Nodes in our DBN model Z_t^k are divided into two sets where t represents the slice number which indicates the time variable, and k is the number of nodes in each slice. The first set contains the hidden state nodes $X_t^n = \{X_t^1, X_t^2, X_t^3, ..., X_t^n\}$, where n represents the number of hidden states in each slice. Hidden states represent immeasurable variables in our model, and these are usually the variables that we aim to gather information about. And the second set is the set of observable nodes $Y_t^m = \{Y_t^1, Y_t^2, Y_t^3, ..., Y_t^m\}$, where m represents the number of observable nodes in each slice. Observable nodes represent variables that can be measured and are completely or partially observable. These are sometimes called evidence nodes. Note that $n + m = k$ in our model.

Each DBN slice contains n hidden variable nodes which represent pilot actions, and k observable and measurable nodes which represent different simulation variables, and these are all observable in our model. The connections between model nodes are set according to actual relationships between the modelled environment variables. And inter slice connections are restricted to hidden nodes. As we mention this we should mention that with the DBN model we set a prior probability for the first slice in the network $P(X_1)$, and we have a state-transition function $P(X_t|X_{t-1})$, and an observation function $P(Y_t|X_t)$. Since we are working with DBN we assume that the model is first-order Markov [7] (i.e. $P(X_t|X_{1:t-1}) = P(X_t|Xt-1)$), and that observations are conditionally first Markov [7] (i.e. $P(Y_t|Y_{t-1}, X_t) = P(Y_t|X_t)$). So from the two previous assumptions we conclude that inter slice relations are limited to hidden states, observable states are only related to their parent states in the same slice.

Now that the model is complete we would like to explain how it is usually used, after the model is built to model different variables in the environment and their relationships, we train the model using the Expectation Maximization algorithm, for the purpose of training we use a number of data sets which usually contain data for all variables (hidden and observable), after training we apply inference techniques to gather the information we need about hidden variables, these techniques [7] include filtering, prediction, classification, control, abduction and smoothing. When performing inference a new data set is used, and this data set contains environment data that we wish to gather information about and this data set usually contains only observable nodes data, and here were anomalies effect can be detected. Our anomaly detection algorithm works using some inference techniques as basic steps to enable it detect data anomalies.

3 Anomaly Detection in a DBN

Anomaly detection is the process of detecting patterns in data that do not conform to the expected normal patterns. Anomalies are also referred to as outliers which Hawkins [9] defines as "an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism". Our approach to anomaly detection is not based on the typical approach which usually focuses on detecting anomalies in general within data of a given model, instead we take another route. When anomalies occur during the prediction or classification process they often have a ripple-like effect on the descendent states in the same slice and consecutive slices. If the anomaly occurs in one slice, its affect is spread to related states in the same slice and to consecutive slices, albeit the effect is shortly lived and soon all values turn back to normal. So the longer the anomaly occurs, the longer and bigger the effect is. In adaptable online learning models if an anomaly continues to occur for a certain period of time, the model will adapt to it and this anomaly will be then considered the norm. During the inference of trained models new data is used, this data could contain some anomalies when compared to the data that was used to train the model. Data could be considered as an anomaly due to its value which does not belong to the range of acceptable values of a given variable. Or it could have a normal value, but it is not normal for this value to occur at that point of time. The second type of anomalies could pass undetected by the experts, and thus effecting descendent states, and if it continued to occur, it could lead to unexpected values when inference is applied to the model. Our algorithm aims to detect this type of data anomaly. During the inference phase, the model is supplied with a data set containing some anomalies. The anomalies are of an acceptable value but do not occur at the expected time, their effect is propagated to related states in the same and succeeding slices. We suppose that we are able to detect these effects on other state/states Z_t at slice t. So our assumption is that we are processing slice t and that all states Z_t of slice t are observable with known state values. Our objective is to go back trough the slices until we can identify in which states Z_{t-k} an anomaly started to occur which have caused the values of future states to be affected and changed. As we mentioned in the previous section, our DBN model is first-order Markov, and observations are conditionally first-order Markov, this leads to the conclusion that hidden states has an affect only on observable states of the same slice and hidden states in the next slice and are only affected by hidden states of the previous slice. We aim to find the node/set of nodes X^i_{t-k} in slice $t-k$ -where k is unknown- that

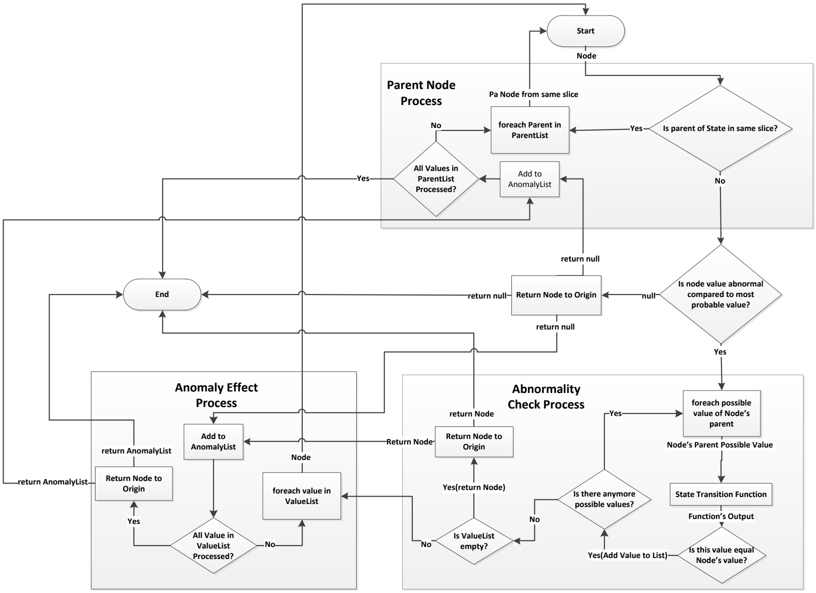

Start
Node
Parent Node Process
Pa Node from same slice
foreach Parent in ParentList
Yes
Is parent of State in same slice?
No
No
All Values in ParentList Processed?
Add to AnomalyList
Yes
return null
End
return null
Return Node to Origin
null
Is node value abnormal compared to most probable value?
return null
Yes
Anomaly Effect Process
Abnormality Check Process
return Node
Yes
foreach possible value of Node's parent
Node's Parent Possible Value
return AnomalyList
Add to AnomalyList
Return Node
Return Node to Origin
Node
foreach value in ValueList
State Transition Function
return AnomalyList
Return Node to Origin
Yes(return Node)
Is there anymore possible values?
Function's Output
No
Yes
All Value in ValueList Processed?
No
Is ValueList empty?
No
Yes(Add Value to List)
Is this value equal Node's value?

Fig. 2. Flow Chart of Anomaly Detection Algorithm

effectively caused a considerable change of value in state Z_t^j in slice t in comparison with the data of the trained model.

As it is apparent from algorithm 1, the algorithm takes as input a state Z_t^j where an abnormal value is detected, and produces as output the state or set of states X_{t-k}^i that had an anomaly which caused this abnormal change of value. At first the algorithm makes sure that the input state does not have any parents in the same slice, if parents do exist this means that the state in concern is an evidence state, then the parent/parents -hidden states- of this state is found. In this case the DETECTANOMALY algorithm is re-implemented on the parent node/nodes. The algorithm calculates the highest probability of any expected value of state Z_t^j at slice t given the trained model. Then this value is compared to probability of the actual value of the state occurring, if there is a large difference between these two values then this data is considered anomalous. Otherwise the algorithm exits. Next step is to go back one slice and to compute the probability of Z_t^j occurring with its current values given all possible values for its parent state and the trained model, this is calculated through the state transition function of the DBN model. If there is a value that supports such transition then state Z_t^j is labelled an affected state, otherwise it is considered as an anomalous state. This process is repeated for all parent states as long as the difference in probability between probable and possible values is above the threshold. When this difference drops below the threshold, the state in that slice is considered normal, and the descendent state in next slice is considered as the first anomalous state in the anomalous path.

Algorithm 1. DetectAnomaly($Z_t^j, Z_t^j.Value$)

```
Require: A State $Z_t^j$ with its Abnormal value.
Ensure: A State or Set of States $X_{t-k}^i$ with Anomalous Data.
if $((Pa(Z_t^j))\&\&(Pa(Z_t^j) \in Slice(t)))$ then
    foreach $(Pa$ in $Pa(Z_t^j))$ do
        AnomalyList·$Add($DetectAnomaly$(Pa, Pa.Value))$
    end for
    return AnomalyList<>
end if
$Z_t^j$.ProbableValue = ComputeMarginals($Z_t^j, Y_t$)
if $(((P(Z_t^j)$.ProbableValue) $- P(Z_t^j$.Value)) >Threshold)&&$((t-1) >= 0))$ then
    $X_{t-1}^i = Pa(Z_t^j) \in$ Slice$(t-1)$
    foreach ($X_{t-1}^i$.PossibleValue in $X_{t-1}^i$.Values) do
        $X_t^i$.ProbableValue = StateTransitionFunction($X_{t-1}^i, X_{t-1}^i$.PossibleValue)
        if ($X_t^i$.ProbableValue == $Z_t^j$.Value) then
            AnomalyEffect = true
            TempStateList·$Add(\{X_{t-1}^i, X_{t-1}^i$.PossibleValue$\})$
        end if
    end for
    if (AnomalyEffect == true) then
        foreach (State in TempStateList<>) do
            TempAnomalyList·$Add$(DetectAnomaly(State,State.Value))
        end for
        if (TempAnomalyList $\neq$ null) then
            AnomalyList·$Add$(TempAnomalyList<>)
        end if
        return AnomalyList<>
    else
        AnomalyList·$Add(Z_t^j)$
    end if
else
    return null
end if
```

4 Experiment and Results

We started by building a DBN model based on a flight scenario, the flight is routed between London Heathrow and East Midlands airports, the flight duration is 50 min on average. We have used Microsoft® Flight Simulator X as the basis for our simulator, because it is supplied with an SDK which was used to build our software, and another reason is that the simulator is very realistic and accurate and it can give us over 1100 different data variables in high frame rates. We have built our custom software that interacts with the simulator and records all of the flight data online with the desired frame rates.

For training purposes we have recorded data of 30 flights between the two designated airports, the flight direction is always the same. 24 of the recorded flights were flown normally with small differences between them. While the remaining 6 flights had data anomalies occurring in them. These anomalies basically where different than normal pilot actions, such as keeping landing gear lowered much more than the usual time, or keeping Flaps at certain angle after takeoff, then we programmed the simulator to cause an effect on the related variables when these anomalies continued to occur, such as having a rough and bouncy landing when landing gears were kept extended longer than they should, which resemble realistic scenarios. The DBN model that we have built is a single layer DBN network, which compromise of two types of nodes, Hidden nodes X_t^n which represent immeasurable pilot actions which are annotated manually into the training data sets, and observable nodes Y_t^m which represent aircraft instrumentation data recorded by our software. Due to the large number of available Sim variables, we had to narrow down the numbers of variables. We have chosen variables which are essential and related to our experiment (i.e. weather data, gps data, altitude and speed data, landing gear data, flaps data, rudder data and data of all cockpit switches that were used during the flight, etc...).

We train the DBN using the Expectation Maximization algorithm in [7], in our training sets we have introduced three types of errors (Landing Gear error, Flaps error, Excess Speed error), each one occurring twice, and the remaining 24 training sets there were no errors. Each one of the errors we have introduced has its own effect. Our algorithm starts working on the slice where the effect appears rather than the slice where the error begins.

In the testing phase we record 9 new data sets with the same types of errors we have introduced, with each error type occurring in 3 different data sets. Note that these data sets do not contain annotated pilot actions, therefore when the algorithm begins data of the observable variables are fetched from the testing data set, whilst data of unobservable variables are entered manually through an annotation step done before running the algorithm.

We start by training the network on 4 training sets, 3 are of normal type and 1 containing an error (Landing Gear error), we run the algorithm on the first of three (Landing Gear error) testing data sets. The algorithm detects 80.7% of the anomaly list after first run, we increase the number of training sets to 9 sets, 8 of which are normal and 1 containing an error, the algorithm detects 82.6% of the anomaly list. We continue adding normal training sets to the training step and the results increase slightly until algorithm reaches 89.4% detection of the complete anomaly list. We repeat the same process but using two different error data sets rather than one. A surprising result was that the algorithm came out with a low detection result (around 57%) when we ran 4 normal data sets with 2 error data sets, we found that the reason for this happing was that we have set a very low value threshold and that the number of error data sets was considered large compared to the number of normal data sets. We have fixed this through increasing the threshold for big error to normal ratios and then increase the ratio of normal to error training sets. After using all 24 normal training sets and 2

Table 1. Error Testing Results

Learnt DS	Lnd Gr Err (1-2)	Flp err(1-2)	exs spd err(1-2)
4	80.1%-82.7%	75.1%-77.7%	80.9%-84.6%
9	82.9%-84.9%	77.9%-79.9%	82.7%-86.4%
14	84.2%-86.4%	79.2%-82.6%	85.9%-88.8%
19	86.6%-88.5%	83.0%-84.5%	88.8%-91.2%
24	88.2%-90.8%	85.2%-87.3%	91.1%-93.6%

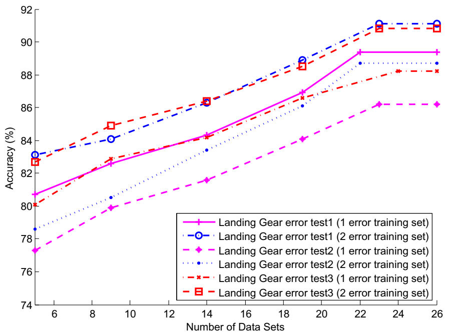


Fig. 3. Landing Gear Error Results

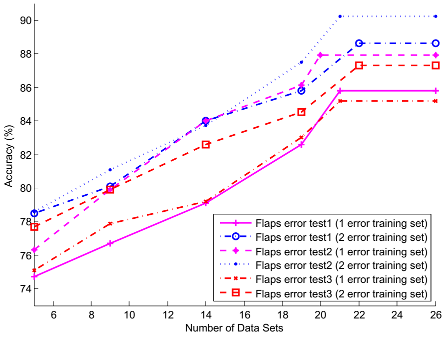


Fig. 4. Flaps Error Results

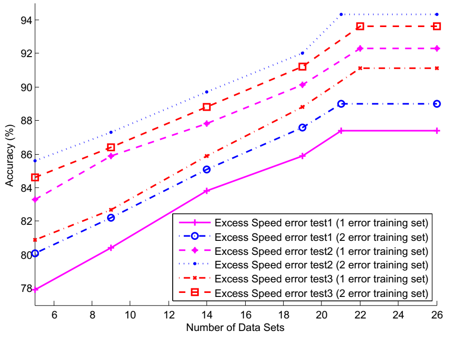


Fig. 5. Excess Speed Error Results

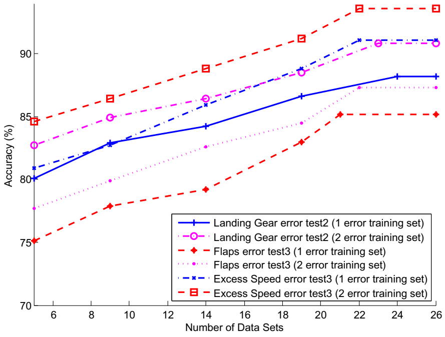


Fig. 6. Error Results Combined

error sets we have reached an anomaly list detection rate of 91.1%. Now we do the same experiment again but on (Landing Gear error) testing data sets 2 and 3 separately. We get final results of 88.7% and 90.8% respectively. We find that the combined accuracy for anomaly list detection with the Landing Gear error is 90.2%, which is considered a very good result. We repeat the same experiment with the remaining two types of errors separately (Flaps error and Excess Speed error) and we get a combined accuracy results of 88.7% and 92.3% respectively. Therefore overall we get a anomaly detection combined accuracy rate of 90.5% with a confidence range of ±1.8%.

5 Conclusion

In this paper we focus on detecting data anomalies in a Dynamic Bayesian Network model, we proposed a novel algorithm to detect data anomalies through backtracking steps of its effect on descendent states until a data anomaly is reached and detected, we have built a DBN model based on pilot actions and instrument data of a flight scenario and then we have implemented our algorithm which has shown robustness in detecting data anomalies that effect other states in the model. Further work can concentrate on the distinction between anomaly types and research its effect on other variables in the model.

References

1. Hodge, V.J., Austin, J.: A survey of outlier detection methodologies. Artificial Intelligence Review 22, 85–126 (2004)
2. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys 41, 15:1–15:58 (2009)
3. Hill, D.J., Minsker, B.S., Amir, E.: Real-time bayesian anomaly detection for environmental sensor data. In: Proceedings of the 32nd Conference of the International Association of Hydraulic Engineering and Research, Venice, Italy (July 2007)
4. Shotwell, M.S., Slatey, E.H.: Bayesian outlier detection with dirichlet process mixtures. Bayesian Analysis 4, 665–690 (2011)
5. Babbar, S., Chawla, S.: On bayesian network and outlier detection. In: Proceedings of the 16th International Conference on Management of Data, Nagpur, India (December 2010)
6. Ferdousi, Z., Maeda, A.: Unsupervised outlier detection in time series data. In: Proceedings of the 22nd International Conference on Data Engineering Workshops, Atlanta, GA, USA (April 2006)
7. Murphy, K.P.: Dynamic Bayesian Networks: Representation, Inference and Learning. PhD thesis, University of California, Berkeley (2002)
8. Pearl, J., Russell, S.: Bayesian networks. In: Arbib, M.A. (ed.) The Handbook of Brain Theory and Neural Networks, 2nd edn. MIT Press (2003)
9. Hawkins, D.M.: Identification of Outliers. Chapman and Hall (1980)

A Novel Road Traffic Sign Detection and Recognition Approach by Introducing CCM and LESH

Usman Zakir, Asima Usman, and Amir Hussain

COSPIRA laboratory, Division of Computing Science, School of Natural Sciences
University of Stirling, Stirling FK9 4LA
usmanzakir@gmail.com, au@instantclaims.org.uk,
a.hussain@cs.stir.ac.uk

Abstract. A real time road sign detection and recognition system can provide an additional level of driver assistance leading to an improved safety to passengers, road users and other vehicles. Such Advanced Driver Assistance Systems (ADAS) can be used to alert a driver about the presence of a road sign by reducing the risky situation during distraction, fatigue and in the presence of poor driving conditions. This paper is divided into two parts: Detection and Recognition. The detection part includes a novel Combined Colour Model (CCM) for the accurate and robust road sign colour segmentation from video stream. It is complemented by a novel approach to road sign recognition which is based on Local Energy based Shape Histogram (LESH). Experimental results and a detailed analysis to prove the effectiveness of the proposed vision system are provided. An accuracy rate of above 97.5% is recorded.

Keywords: Colour Segmentation, Detection, Recognition, CCM, LESH, SVM, ADAS.

1 Introduction

Road signs have meanings depending on their colours, shapes used and contents included within. Primarily, road sign colours are Red, Blue, Green, Brown, Yellow or White, which signify and categorize their importance, e.g. Red for obligatory signs and Blue for advisory signs. Therefore, colour plays an important initial role in a typical road sign detection task. Similarly, the global and local shape related features of a road sign can provide important clues in distinguishing one sign from another, i.e. in the recognition of the detected road signs. Due to varying lighting and weather conditions, segmentation of road signs using colour information and their recognition based on shape features; especially in outdoor images is a significantly challenging task. A detailed literature review carried out by the authors on automatic road sign detection and recognition revealed that even though a significant amount of research has been carried out in Road Sign Detection and Recognition (RSDR), none of the published work has either considered the use or all available colour representation schemes in colour based segmentation nor attempted to provide a detail comparison

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 629–636, 2012.

as to how different colour spaces perform under changes of illumination and weather conditions. Further the possibility of use of Local Energy based Shape Histograms (LESH) has not been investigated before in the context of road sign recognition. This research is an attempt to bridge this research gap with the ultimate aim of recommending an efficient and robust colour model to be used in automatic road sign detection and an efficient invariant shape based features for recognition of road signs, under varying environmental conditions. For clarity of presentation this paper has been organized as follows: In addition to this section in which the research problem was introduced and its practical relevance was highlighted, Section-2 provides exiting State of the Art, Section-3 presents the proposed approach providing details about each operational stage. Section-4 provides the experimental results obtained and an analysis of the results leading to the conclusions that are provided in Section-5.

2 State of the Art

This section introduces existing literature in the application domain of the RSDR. The study of these approaches will conceptually compare the performance of the state of the art to the performance of the proposed approaches. This allows fair comparison, particularly when no standard dataset is available for researchers to carry out performance analysis. Road sign detection from an image or image sequence is the first key step of the RSDR. An extensive investigation of existing literature [1] , [2] and [3] has been made which reflects that using the properties of colour; shape or joint information carry out the detection step. Secondly road sign recognition is performed on the contents of the candidate road signs and is mostly dependent on an extensive shape analysis and classification. In addition, road sign tracking is adopted by some researchers to enhance the accuracy of the detection and recognition stages and to reduce the computational cost of having not to repeat the above processes on each video frames. A detailed taxonomy is provided in [1] shows varying ranges and combinations of the algorithms utilised by the RSDR systems in the literature. Colour based segmentation is achieved by using different colour models; Shape is also considered as an important feature of the road sign representation and Contents are recognised by utilizing various feature extraction techniques and classifiers. The next section aims to overcome previous research gaps in designing a robust road sign detection and recognition system that is capable of performing on video streams under wide variations of illumination and environmental conditions.

3 Proposed Approach

A number of key stages constitute the complete road sign detection and recognition system as shown below see Fig 1- “Road sign detection and Recognition Framework”.

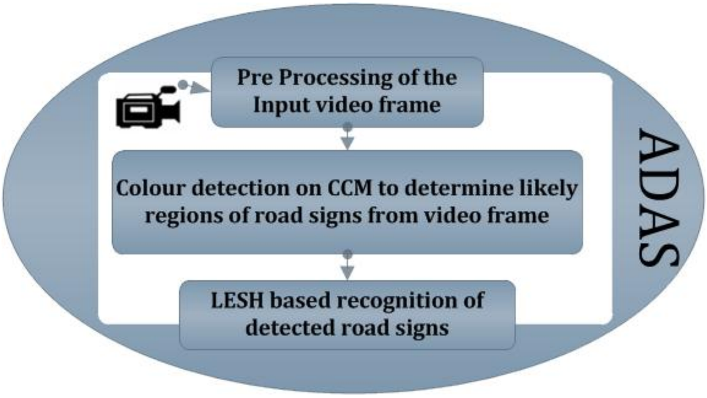

Pre Processing of the Input video frame
Colour detection on CCM to determine likely regions of road signs from video frame
LESH based recognition of detected road signs
ADAS

Fig. 1. Road sign detection and Recognition Framework

3.1 Pre- Processing

In this stage each captured video frames from a video camera are partitioned horizontally at a ratio 7:3 (i.e. (7) top: (3) bottom) and the bottom third of the video frame can be ignored in the subsequent processing. This is based on the assumption that the camera is mounted at a view position of the car driver and the bottom third of the image will be mostly consisting of front bonnet of the car or road surface as seen through the windscreen by the car operator.

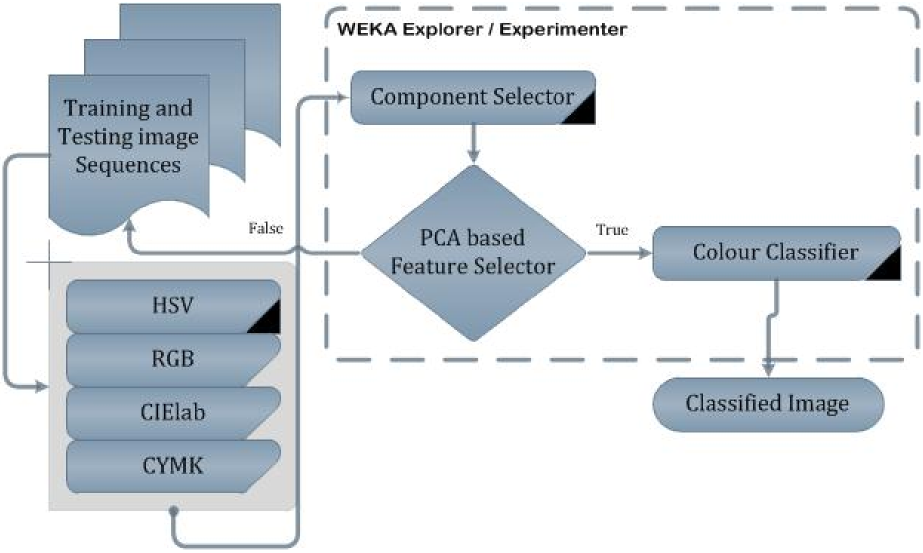

Training and Testing image Sequences
WEKA Explorer / Experimenter
Component Selector
PCA based Feature Selector
False
True
Colour Classifier
Classified Image
HSV
RGB
CIElab
CYMK

Fig. 2. Colour classification based on CCM

3.2 CCM Based Detection

The idea of CCM focuses on the merger of the properties of four distinct colour spaces in the presence of wider range of illumination variance which makes the detection task further robust. This enhances the colour segmentation accuracy of the road signs where single colour space based segmentation fails to extract the desired colour information. Fig. 2-"Colour classification based on CCM", shows the block diagram of CCM which has been explained in this section. The model initiates with the retrieval of equivalent colour pixel values from four colour spaces i.e HSV, RGB, CIElab, and CYMK. The training images are the samples of road sign colours captured in varying illumination, weather and scaling conditions. In this model we have only obtained three distinct colour samples of road signs i.e. Red, Green and Blue. It is assumed that the training images are represented by RGB colour space. The gamma values are decoded prior to the transformation of these images to other three colour spaces i.e. HSV, CIElab and CYMK. The transformation of RGB images to the above mentioned colour spaces is detailed in [1]. The pixel information is obtained from each manually achieved training colour sample of road sign. Red, Green and Blue colour pixels are represented by 13 *(3 components each from HSV, RGB and CIElab whereas 4 components of CYMK)* components in total for each colour. The 13 components vector representing one particular colour pixel from all colour spaces can be picked at any random order. This has to be noted that each component value of colour pixel were obtained manually from a sample colour of a road sign captured in various illumination conditions. Our experiments revealed that the use of a 13 dimensional feature vector to represent a single colour pixel leads to an unacceptable level of computational cost in the classification.

Table 1. Results of Search methods with selected components and Quantity

Component Selector	Selected Components	Number of Components
Best First	H,R,B,a,b,C,M	7
Exhaustive Search	H,R,B,a,b,C,M	7
Genetic Search	H,R,B,a,b,C,M	7
Greedy Stepwise	H,A,b,C,M,K	6
Random Search	H,R,G,B,a,b,C,M	8

Thus in order to reduce colour components set and the computational complexity in the subsequent stages or to reduce the chances of data over-fitting, component selectors of WEKA package is introduced. To obtain the optimum set of selected components combination, it is investigated that the use of five different popular search methods namely, Best First, Exhaustive Search, Genetic Search, Greedy Stepwise Search and Random Search. It was revealed that the set consisting of 7 components, H, R, B, a, b, C and M (*Hue, Red, Blue, a and b chroma components,* *Cyan and Magenta respectively*) were selected on an average to be the most appropriate components to represent a pixel colour value as shown in the Table 1. Thus the components set have been reduced from an original set of 13 components to 7 components. This removes any redundancy present between components by disregarding components which are non-significant in data discrimination. The selected components are further analysed by using Principal Component Analysis (PCA). Each colour class i.e. Red, Green and Blue is converted into feature space in this analysis. This further helps in reducing the data dimensionality and redundancy. Table 2 shows the Eigen Vectors obtained against 7 components for each class respectively. The Eigen Vectors for Red Class are obtained from its 110 colour pixel instances. Similarly Eigen Vectors for Green and Blue Classes are obtained from their 99 colour pixel instances respectively. The data transformation from component representation to feature space causes the dimensionality reduction. That is the components representing a particular colour pixel with 7 dimensions, are represented with 1 dimension in feature space. Each feature represents a colour pixel that carries a unique instance within its designated class. These features are later trained on SVM multiclass polynomial kernel for the classification of colour pixels from the input test image. The classified image represented in binary image format, where classified pixels are represented with white and non-classified are represented as black pixels. The next section explains about recognition of road signs which were detected with the method described in this section.

Table 2. Eigen vectors for RED, BLUE and GREEN colours

	V1	**V2**	**V3**	**V4**	**V5**	**V6**	**V7**	**V8**	**V9**
	Red Colour Eigen Vectors			**Blue Colour Eigen Vectors**			**Green Colour Eigen Vectors**		
H	-0.0902	-0.0334	-0.7159	0.0406	0.4721	0.6379	0.0484	0.2146	0.7632
R	-0.4869	0.3552	0.1	-0.4559	0.1798	0.3904	-0.4763	0.2148	-0.3829
B	-0.5188	-0.2422	-0.1549	-0.3896	0.3487	-0.3967	-0.4358	0.3862	0.242
A	-0.0367	0.6121	-0.3614	0.2699	0.5755	0.0087	0.1893	0.5218	-0.3848
B	0.2534	0.483	0.4061	-0.1774	-0.5296	0.47	-0.0877	-0.6323	-0.1382
C	0.484	-0.3606	-0.0955	0.5222	-0.1066	-0.0809	0.5748	-0.0458	0.0672
M	0.431	0.2757	-0.3856	0.5115	0.0098	0.2362	0.455	0.2908	-0.2015

3.3 Recognition

Once the classified binary image is obtained, it initiates the process of recognition and classification of the road sign contents. The recognition process comprises of the LESH [2], [5] features extraction of the road sign contents and training/testing of these features by employing SVM polynomial kernel. The candidate road signs, which are validated in the detection module, are further processed to obtain the valid road sign contents for feature extraction. The internal contents of road signs are normally represented as black and white colours. The white and black areas can be extracted by simple black and white region extraction using adaptive threshold. After obtaining the binary images, the connected components from the binary image are extracted this removes the noisy objects (non-sign objects) at the same time. The image(s) are normalised to a square dimensional image of size 128×128 and at the same time converted to grey level image. It should be reminded that the image normalisation to a fixed dimensional size and its grey level conversion are the valid input requirements for LESH feature extraction. LESH features are obtained by computing the local energy along each filter orientation of image sub-region. The overall histogram represents the concatenated histograms, which are computed along each sub-region of the image. These extracted LESH features from different classes of road signs are trained and classified with the help of multiclass SVM polynomial kernel.

4 Experimental Setup and Results

This section provides experiments carried out on the video samples of miscellaneous road signs captured during varying lighting conditions. The resolution 640×480 pixels is used to capture testing video samples whereas 2592×1944 pixels resolution is used to capture images for training purposes. The hardware comprises of *Canon IXUS80IS* digital camera for image and video acquisition, Pentium 4 Dual Core 3.2 Ghz, and 4 GB of RAM. The RSDR application is developed and tested by using Visual Studio .Net and signal and image processing toolboxes of MATLAB.

Table 3 presents a set of miscellaneous road signs group (e.g. Advisory and Obligatory etc.) i.e. '*Round About*', '*Stop*', '*Slippery Road*, '*Speed 30'* and 'Give way ', and they are given class labels for this experiment as *T1, T2, T3, T4 and T5* respectively. The training of these road signs is performed on 40 image samples per class. The testing is performed on the video samples of road signs captured during poor weather conditions, partial occlusion and abnormal orientation. The confusion matrix of *miscellaneous road signs* is presented in Table 3. The ROC (Receiver Operating Characteristic) curve for tested *miscellaneous* road signs is presented in Fig.3.-"ROC curve of miscellaneous road signs", where true positives are plotted against false positives. This has to be noted that the set of these miscellaneous signs have underperformed when tested under single colour space based segmentation and LESH based recognition [3].

Table 3. Confusion Matrix of tested miscellaneous road signs

Road Signs	**True Labels**	**Estimated Labels**					**Totals**
		1	**2**	**3**	**4**	**5**	
	T1	18	1	0	1	0	**20**
	T2	1	28	0	0	0	**29**
	T3	1	0	52	0	1	**54**
	T4	0	0	0	64	0	**64**
	T5	0	0	0	0	56	**56**
	Totals	**20**	**29**	**52**	**65**	**57**	**223**

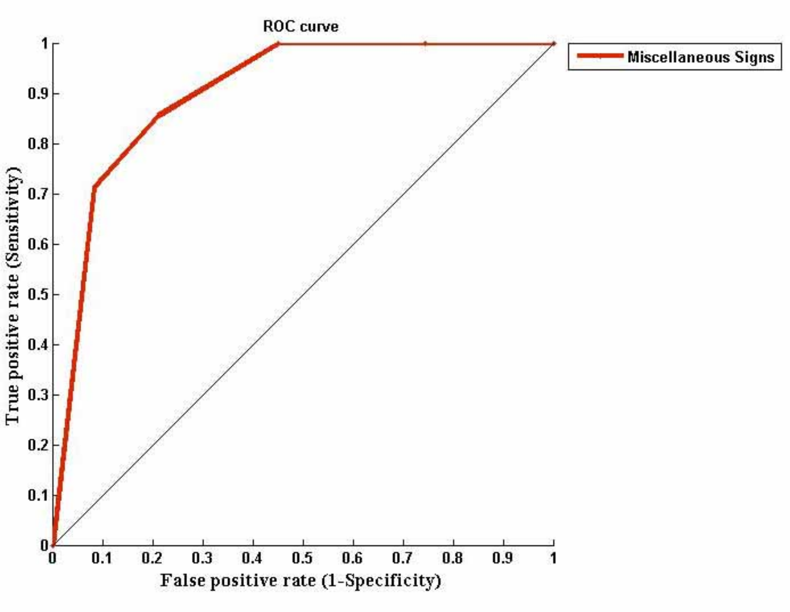


Fig. 3. ROC curve of miscellaneous road signs

5 Conclusion

In this paper we have presented a robust approach to real-time road sign detection. The algorithm utilizes a novel combined colour model for accurate detection of road sign from video stream which is proven to provide accurate results under significant variations of scene illumination and the presence of different ambient light source types. The combined colour model is the combination of properties of HSV, RGB, CIElab and CYMK colour spaces. The training process of this model initiates from obtaining the colour pixels information from the road signs; captured at various ambient levels. The equivalent colour pixel values are obtained for HSV, CIElab and CYMK colour spaces. The dominant components of these colour spaces are extracted with average results of popular search methods i.e. Exhaustive Search, Basic Search, Genetic Search, Greedy Stepwise and Random Search. The selected components representing a particular colour pixel are transformed to colour features by using PCA filter. The colour features are trained as three colour classes i.e. Red, Green and Blue using SVM multiclass classifier. The test image is classified with the help of colour classifier to extract the candidate sign for further analysis. The Recognition stage introduces the SVM classifier with the Local Energy based Shape Histogram (LESH) features. Overall accuracy figures of 97-99% have been reported. We are currently working on real time application of the algorithm within an in-car navigation system.

References

1. Automatic Road Sign Detection and Recognition, PhD Thesis, Computer Science Loughborough University (2011), `http://lboro.academia.edu/usmanzakir/Papers/1587192/Automatic_Road_Sign_Detection_And_Recognition`
2. Zakir, U., Zafar, I., Edirisinghe, A.E.: Road Sign Detection and Recognition by using Local Energy Based Shape Histogram (LESH). International Journal of Image Processing 4(6), 566–582 (2011)
3. Zakir, U., Edirishinghe, E.A., Hussain, A.: Road Sign Detection and Recognition from Video Stream Using HSV, Contourlet Transform and Local Energy Based Shape Histogram. In: Zhang, H., Hussain, A., Liu, D., Wang, Z. (eds.) BICS 2012. LNCS, vol. 7366, pp. 411–419. Springer, Heidelberg (2012)
4. Zakir, U., Leonce, J.N.A., Edirisinghe, A.E.: Road sign segmentation based on colour spaces: A Comparative Study. In: Proceedings of the 11th Iasted International Conference on Computer Graphics and Imgaing, Innsbruck, Austria (2010)
5. Sarfraz, S.M., Hellwich, O.: An Efficient Front-end Facial Pose Estimation System for Face Recognition. International Journal of Pattern Recognition and Image Analysis 18(3), 434–441 (2008)
6. Do, N.M., Vetterli, M.: The Contourlet Transform: An Efficient Directional Multi resolution Image Representation. IEEE Transactions on Image Processing 14, 2091–2106 (2005)

On the Application of Bio-inspired Algorithms in Timetabling Problem

Daniela Oliveira Francisco, and Ivan Nunes da Silva

University of São Paulo, Department of Electrical Engineering, CP 359
CEP 13566.590, São Carlos, SP, Brazil
{doliveira,insilva}@sc.usp.br

Abstract. Timetabling is a classical problem discussed extensively in the literature due to the widespread need for quality timetables. Most educational institutions still prepare their timetables manually, which is a highly time-consuming process and subject to errors. Several approaches to solve this problem are also found in technical studies, which use stochastic search methods due to the problem's complexity. The optimization strategies formulated and compared in this study are based on genetic algorithms and artificial immune systems. The proposed techniques provide quality solutions for the timetabling problem.

Keywords: Genetic Algorithms, Artificial Immune Systems, Timetabling Problem, Systems Optimization.

1 Introduction

The generation of quality timetables is a critical factor in any educational institution. This is considered a complex problem because it involves several types of information, such as schedules, course subjects, teachers, students, etc., according to Pillay and Banzhaf [1]. Several search strategies have been applied to solve timetabling problems, whose constraints may vary from one educational institution to another [2].

Wang, Liu and Yu [3] state that the generation of timetabling must satisfy the constraints imposed by the institutions on which the problem is based. The more constraints the solution satisfies, the better it will be adapted to the problem.

The timetabling optimization systems proposed here are systematic and automated procedures that generate tables containing all the subjects of a given course, organized by semester, and which also consider the available resources.

The simulated results of two search and optimization methods were applied and compared in this study, i.e., genetic algorithms (GA) and artificial immune systems (AIS). GA and AIS are evolutionary algorithms inspired by biological metaphors. In this case, the GAs are based on Darwin's theory of evolution, while the AISs are based on the natural immune system [5]. Decision support systems, which are responsible for automatically generating timetables, were developed based on GA and AIS, taking into account the most common constraints reported in the literature.

This paper is divided into six sections as follows. Section 2 describes the genetic algorithms that were applied in this research and how they work. Section 3 discusses

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 637–644, 2012.

artificial immune systems, highlighting the functions of the clonal selection algorithm also used in this study. Section 4 describes the timetabling problem, the algorithms that were developed, and the parameters that were used for their configurations. Section 5 describes the results of the applications developed here, and lastly, Section 6 offers our conclusions and the main contributions of this work.

2 Genetic Algorithms

According to Goldberg, Korb and Deb [4], genetic algorithms are stochastic search techniques based on Darwinism and on concepts of nature genetics. Computational imitation and simulation of natural processes produces very interesting results.

In any given population, individuals with superior genetic characteristics are more likely to survive and to produce increasingly fit individuals, while less fit individuals tend to disappear from the population.

Given any random optimization problem, genetic algorithms search for a response based on a random set of solutions. Each of these solutions is called an individual or chromosome. An individual represents a complete solution to the problem in question. Thus, genetic algorithms favor the combination of the fittest individuals, or the most promising ones for the solution of a given problem, working with set of encoding parameters and not with their own parameters.

During the evolutionary process, the population is evaluated as follows: each individual receives a fitness score, which indicates its ability to adapt to a particular environment. The natural selection process is simulated using the fittest individuals.

Genetic operators are applied to selected individuals, thereby generating new individuals. New populations are generated until the stop condition is satisfied. A stop condition can be defined by specifying a maximum number of generations, or when a satisfactory solution to the problem has been reached.

Additionally, genetic algorithms operate in parallel on a population of candidate solutions. Searches are made in different areas of the solution space, allocating an appropriate number of members to search in several regions. Thus, this technique has a greater chance of reaching the most promising areas of the search space because it works with a population of solutions rather than a single point.

A chromosome has its own genotype, representing the encoding of the solution, and its phenotype, which represents a possible solution to the problem. Usually, chromosomes are lists of attributes or vectors, in which each attribute is known as a gene and its possible values are called alleles.

Genetic algorithms belong to the class of probabilistic algorithms, but they are not purely random search methods, since they combine direct search and stochastic methods.

Genetic algorithms have proved to be efficient in finding optimal (or satisfactory) solutions for a large class of optimization problems because, unlike traditional methods, they do not involve many constraints. Although this technique may seem simplistic in comparison to natural biological structures, it is sufficiently complex to provide robust adaptive search mechanisms.

3 Artificial Immune System

Artificial immune systems are part of the research area inspired by natural systems, known as biologically inspired systems. The purpose of bio-inspired systems is to computationally model the mechanisms found in nature. Artificial immune systems belong to the class of bio-inspired algorithms based on the natural immune system.

The natural immune system is responsible for protecting organisms from pathogenic agents. Throughout an individual's life, his immune system adapts continually to recognize harmful agents and respond effectively when under an attack from those pathogens. This process of adaptation enables the individual to develop immune memory cells, which form the immune defense system. The ability of the immune system to adapt during its first exposure to an antigen, and to create specific antibodies to generate an immune response, serves as the basis for the theory of clonal selection.

The Clonal Selection Algorithm (CLONALG) proposed by He, Hui and Lai [5] is based on the principle of clonal selection. The CLONALG involves the steps of initializing the population, the clonal expansion process, and variation of the population, as follows:

- Random generation of the initial population of antibodies. Each antibody in the population represents a solution that is completely relevant to the problem in question;
- Evaluation of the population by means of the objective function, which determines the affinity of each antibody;
- Through the process of selection by affinity, the *n* best antibodies of the population will be chosen and subjected to clonal expansion. The process of clonal expansion consists of cloning the chosen antibodies and maturation of the clones in order to improve the solutions;
- Each new antibody generated by clonal expansion will be evaluated and its affinity compared with the affinity of the original antibody. The antibody with the highest representativeness will then be inserted into the new population;
- Finally, the diversity will be inserted into the population in order to prevent the algorithm from converging prematurely: the worst *w* antibodies will be replaced by new *w* antibodies;
- The evaluation procedure, clonal selection, clonal expansion and insertion of genetic diversity are repeated until the predefined stop criterion is satisfied.

The above described algorithm enables local searches by maturing the clones, as well as global searches by inserting diversity into the population.

According to Castro and Von Zuben [6], the solution to the problem is obtained from solutions adapted during the evolutionary process of the CLONALG, which is inherent to the natural immune memory and is enhanced throughout an individual's life.

4 Application of Bio-inspired Algorithms to Timetabling Problems

According to Yue, Li and Xiao [7], the problem of timetabling optimization is considered a NP-complete problem due to its mathematical complexity. Thus, it requires the application of non-deterministic and stochastic search methods.

The search optimization methods used in this work to solve the timetabling problem are genetic algorithms and the clonal selection algorithm, whose satisfactory results when applied to optimization problems are reported in the literature.

Two decision support systems were developed in this work, combining heuristic techniques with the genetic algorithms and the clonal selection algorithm. The purpose of this research is to make a comparative analysis of the two techniques in order to determine which one offers the most promising results for solving the timetabling problem.

This problem has characteristics and constraints that may vary according to the educational institution for which the implementation is intended. High-level constraints were also adopted here, i.e., if any of these constraints is violated, the results will be invalid.

The software programs developed here are systematic and automated procedures for generating timetables, containing all the subjects of a course, organized by semester, considering the availability of the following resources:

- The disciplines taught in a semester cannot be allocated at the same time;
- The courses taught by a teacher cannot be allocated at the same time;
- The availability of each teacher must be checked, and the subjects he teaches cannot be allocated at the times when he is not available.

In this paper, the encoding of each chromosome (in GA) or antibody (in CLONALG) represents a complete timetable containing all the subjects of a course, organized according to the semester in which they belong.

The initial population in the two algorithms is generated by a computational function that combines a random routine with heuristic techniques. The use of heuristics techniques is justified by the good values of fitness obtained in GA and of affinity in CLONALG, and by the computational effort involved in achieving the convergence of the algorithms. The heuristic techniques adopted here tend to generate a better adapted population, which includes the time slots when the teacher is absent from the university and should therefore not be allocated, thus satisfying one of the constraints of the problem upon the initialization of the population.

The objective function, which determines fitness (in GA) or affinity (in CLONALG), applied in the evaluation of each candidate seeks to find a feasible solution that satisfies all the constraints. In this work, fitness and affinity were defined as a counter that is incremented each time a solution has a feature that violates any of the constraints. Whenever a resource constraint is violated, the fitness (or affinity) must necessarily be increased.

The stop condition applied here is the moment when the result of the objective function reaches a value equal to zero, i.e., when the solution that is obtained is considered viable because none of the constraints has been violated and a feasible timetable has been generated.

The selection methods employed here to select the solutions for the next generation, as well as for the application of genetic operators to GA and of clonal expansion to CLONALG, were the methods that search for the best solutions for the problem in question, i.e., the rank selection method (for GA) and the affinity selection method (for CLONALG). In both of these selection methods, the *n* solutions with the best values obtained through the objective function will be chosen.

The different parameters of each technique adopted in this study are described below.

4.1 Genetic Operators Applied to GA: Mutation and Crossover

Genetic operators are used in GA to insert diversity into the population and achieve the goal of this research, i.e., that of devising a feasible timetable. The genetic operators used here were mutation and crossover.

The purpose of mutation is to insert a small measure of diversity into the population. Thus, it is advisable to use a low mutation rate to avoid losing the advances that have been achieved through the use of the heuristic methods in the initialization of the population. The steps involved in the mutation were as follows:

- Define the mutation rate;
- Randomly choose the individuals to be mutated;
- Randomly select two different points among the selected individuals;
- Change all the selected points in the selected individuals, in each semester;
- Evaluate the individuals thus generated.

The mutation adopted here does not violate the constraints, since the change is applied in all the semesters of the solution, at the same points.

The crossover operator performs the crossover between two randomly selected chromosomes, thus generating a new chromosome that possesses characteristics of the two original chromosomes. The newly generated chromosome must be checked and, if necessary, also restructured, since it may violate all the constraints imposed by the problem. The steps involved in the crossover were as follows:

- Select the crossover rate;
- Randomly select individuals to which the crossover will be applied;
- Select a cutoff that corresponds to a time slot;
- To generate the new chromosome, the genes of a chromosome will be used up to the cut-off point. The remaining genes will be extracted from the next selected chromosome, starting at the cut-off point, until the end of the timetable is reached;
- Check if all the course subjects have been allocated in the new chromosome;
- If necessary, correct the new chromosome to ensure that all the course subjects are included in the proper semester, with the correct workload;
- Evaluate the new individuals thus generated.

4.2 Clonal Expansion of CLONALG

The clonal expansion process in CLONALG involves the following steps: cloning of the *n* best antibodies, maturation of the clones, and evaluation of the clones.

The number of clones was determined by the following equation:

$$N_c = \sum_{i=1}^{n} round\left(\frac{\beta.N}{i}\right) \qquad (1)$$

where N_c is the total number of generated clones; β represents the multiplication factor, defined here as $\beta = 1$; *N* represents the number of antibodies in the population, and *round* is the operator responsible for transforming the numerical result into an integer value [6].

Maturation corresponds to the mutation applied to the clones, whose rate is calculated inversely proportional to its affinity. The maturation adopted here consists of the following steps:

- Randomly select different points in the selected and cloned timetables, in order to change these time slots;
- Change all the selected points in each clone, in all the semesters;
- Evaluate the newly generated antibodies;
- Perform a comparative analysis of the affinity values to check if the representativeness of the new antibody in the population is greater than that of the antibody from which it originated.

After concluding the maturation process, the timetables are evaluated and the antibodies with the highest affinity are inserted into the subpopulation.

4.3 Inclusion of Diversity in the Population of CLONALG

The process of inserting diversity into populations consists of substituting the timetables with the lowest affinity values for new solutions, which are generated by a function that randomly determines the codes of the solutions, in combination with the heuristic techniques, as was done to generate the initial population of this algorithm.

5 Results of the Experiments

The results of the execution of the decision support systems developed here are shown below. Since our goal was to obtain valid timetables at the lowest possible computational cost, several computer simulations were performed to determine the ability of each of the algorithms to adequately explore the search space of the problem.

Table 1 lists the results of thirty runs of the clonal selection algorithm and thirty runs of the genetic algorithms in each configuration adopted for the genetic operators of crossover and mutation. Several tests were conducted to determine the most suitable crossover and mutation rates for this application.

Table 1. Experimentals results

Algorithms	Number of Software Runs	Number of Iterations	Mutation Rate (%)	Crossover Rate (%)
CLONALG	*30*	*13953*	-	-
AG	*30*	*17865*	*10*	*80*
	30	*24630*	*3*	*30*
	30	*19562*	*5*	*60*

As Table 1 indicates, CLONALG was run thirty times, which means that thirty feasible timetables were created, since the purpose of the stop condition adopted for this problem was to find timetables in which the predefined constraints were not violated. The third column in Table 1 shows the number of iterations required to create thirty feasible timetables. The same stop condition was adopted for the GA, and as indicated in Table 1, thirty valid timetables were generated in each of the adopted configurations.

The most suitable configurations for GA were found at a mutation rate of 10% and a crossover rate of 80%, as indicated by the results in Table 1, lines 2, 3 and 4. A comparison of GA and CLONALG confirmed the superiority of CLONALG, which required fewer iterations to produce the same number of feasible timetables as those produced by the GA, even in the best configuration of GA.

6 Conclusions

The purpose of this work was to perform a comparative analysis of the results obtained with genetic algorithms and artificial immune systems when applied to solve the timetabling problem in educational institutions. The main contribution of this work is the development of decision support systems that are responsible for the automated generation of feasible timetables, which were based on the most common parameters adopted by universities.

As the results in Table 1 indicate, the clonal selection algorithm produced better results than the genetic algorithms, considering the parameters adopted here, since it required fewer iterations to obtain a valid timetables. The adoption of appropriate heuristic techniques to generate the initial population was a factor of success in this study, for it enabled faster convergence of both the algorithms developed here, and thus the identification of a feasible solution at low computational cost.

Proposals for future research will involve the discussion and analysis of other bio-inspired algorithms described in the specialized literature, such as Artificial Endocrine Systems and Particle Swarm Optimization, as well as their development based on parallel programming, in order to perform algorithm workload distribution on multiple processors and find solutions for the problem in the shortest possible time.

References

1. Pillay, N., Banzhaf, W.: An Informed Genetic Algorithm for the Examination Timetabling Problem. Applied Soft Computing 10, 457–467 (2010)
2. Suyanto, S.: An Informed Genetic Algorithm for University Course and Student Timetabling Problems. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 229–236. Springer, Heidelberg (2010)
3. Wang, Z., Liu, J.L., Yu, X.: Self-Fertilization Based Genetic Algorithm for University Timetabling Problem. In: 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation, pp. 1001–1004. ACM, New York (2009)
4. Goldberg, D.E., Korb, B., Deb, K.: Messy Genetic Algorithms: Motivation, Analysis, and First Results. Complex Systems 3, 493–530 (1989)
5. He, Y., Hui, S.C., Lai, E.M.-K.: Automatic Timetabling Using Artificial Immune System. In: Megiddo, N., Xu, Y., Zhu, B. (eds.) AAIM 2005. LNCS, vol. 3521, pp. 55–65. Springer, Heidelberg (2005)
6. Castro, L.N., Von Zuben, F.J.: Artificial Immune Systems (Part II) – A Survey of Applications. Technical report, FEEC/UNICAMP (2000)
7. Yue, Z., Li, S., Xiao, L.: Solving University Course Timetabling Problems by a Novel Genetic Algorithm Based on Flow. In: Liu, W., Luo, X., Wang, F.L., Lei, J. (eds.) WISM 2009. LNCS, vol. 5854, pp. 214–223. Springer, Heidelberg (2009)

A Bio Inspired Estimation of Distribution Algorithm for Global Optimization

Omar S. Soliman and Aliaa Rassem

Faculty of Computers and Information, Cairo University, 5 Ahmed Zewal Street,
Orman, Giza, Egypt
Dr.omar.soliman@gmail.com
aliaa.rassem@yahoo.com

Abstract. This paper introduces a new bio-inspired Estimation of Distribution Algorithm for global optimization that integrates the quantum computing concepts with the immune clonal selection, vaccination process and Estimation of Distribution Algorithm (EDA). EDA is employed in the vaccination process to improve the solutions diversity and maintain high quality solutions in addition to its ability to avoid falling in local optimum for multi modal problems. The proposed algorithm is implemented and evaluated using standard benchmark test problems. Experimental results are compared with the quantum inspired immune clonal algorithm (QICA) and the QICA- with vaccine algorithm, where the proposed algorithm is superior to both of them. The obtained results carried out, it is performing well in terms of the solutions quality and diversity, and it is superior to both of compared algorithms.

Keywords: Quantum Inspired Immune Clonal Algorithm (QICA), Estimation of Distribution Algorithm (EDA), Vaccine Operator, Global Optimization.

1 Introduction

Immune clonal algorithm (ICA) is inspired from the human immune systems clonal selection process over the B cells where the evolution process of the antibodies is a repeated cycle of matching, cloning, mutating and replacing. The best B cells are allowed through this process to survive which increases the attacking performance against the unknown antigens. Vaccination is another immunological concept that ICA applies through the vaccine operator to introduce some degree of diversity between solutions and increase their fitness values [1], [15].

The quantum-inspired immune clonal algorithm (QICA) is one of the Quantum inspired evolutionary algorithms QIEAs, based on the combination of quantum computing principles, like quantum bits, quantum superposition property and quantum observation process, with immune clonal selection theory. The quantum bit representation for antibodies and vaccines has the advantage of representing a linear superposition of states (classical solutions) in search space probabilistically. Quantum representation can guarantee less population size as

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 645–652, 2012.

a few number of antibodies and vaccines can represent a large set of solutions through the space [16]. The quantum observation process plays a great role in projecting the multi state quantum antibodies into one of its basic states to help in the individuals evaluation. Quantum vaccine ICA algorithm(QICA-V) is an algorithm that applies quantum vaccines to inject the quantum antibodies in the search space to increase their fitness. This algorithm has a drawback in its obtained solutions because they have a lack in diversity. A new hybridization of QICA-V and Estimation of distribution Algorithm is proposed to obtain some degree of diversity between solutions and maintain low computational and time complexity.

The aim of this paper is to develop a bio-inspired algorithm based on the QICA and the vaccine operator with the aid of EDA to sample vaccines. The algorithm merges the quantum computing concepts with the vaccine operator and the EDA sampling to improve the diversity and save computational time. The rest of this paper is organized as follows: Section 2 introduces a brief of some related work that had been done using QIEA and a background about QICA and EDA algorithms. The proposed algorithm is presented in section 3. The experiments setup and results are presented in section 4, where the last section is devoted to conclusions and further researches.

2 Related Works and Background

Quantum-Inspired Artificial Immune System algorithms had been applied extensively in virous real applications [5,7,9,12,17]. The vaccine operator was also used in many works with the AIS algorithms to enhance their exploration ability and increase their detection efficiency [2,6,13,14,18].

2.1 Estimation of Distribution Algorithm

iterated density estimation evolutionary algorithms (IDEAs) are EAs that apply an explicit sampling procedure through using probabilistic models rep- resenting the solutions characteristics. Estimation of Distribution Algorithms (EDAs) are types of the IDEA and population based algorithms with a theoretical foundation of probability theory. They can extract the global statistical information about the search space from the search so far and builds a probability model of promising solutions [1, 4, 9, 15]. The general procedure of EDA is described in algorithm 1.

The EDA advantage is that it relies on the construction and maintenance of a probability model that generates satisfactory solutions for the problem solved. An estimated probabilistic model, to capture the joint probailties between variables, is constructed from selecting the current best solutions and then it is simulated for producing samples to guide the search process and update the induced model. Estimating the joint probability distribution associated with the data constitutes the bottleneck of EDA. Based on the complexity of the model used, EDAs are classified into different categories, without interdependencies,

Algorithm 1. Estimation of Distribution Algorithm

```
Initialize the initial population.
while termination condition is not satisfied do
    Select a certain number of excellent individuals.
    Construct probabilistic model by analyzing information of the selected
    individuals.
    Create new population by sampling new individuals from the constructed
    probabilistic model.
end while
```

pair wise dependencies and multiply dependencies algorithms where detailed description is shown in [15].

2.2 Quantum Inspired Immune Clonal Algorithm

Quantum-Inspired Artificial Immune System algorithms had been applied extensively in virous real applications [5, 7, 9, 12, 17]. The vaccine operator was also used in many works with the AIS algorithms to enhance their exploration ability and increase their detection efficiency [2,6,13,14,18]. Quantum inspired ICA (QICA), is the hybridization between QC and classical ICA to enhance the perfrmonace of the ICA and helpe in solving the problem of its ineffective performance in high dimensional problems. Inspired quantum concepts used in QICA include quantum bit (q-bit), quantum mutation gate and observation process [16].

3 Proposed Algorithm

The proposed algorithm integrates the quantum computing and immune clonal selection principles with the vaccination and EDA sampling mechansim to improve the solutions fitness and degree of diversity. The quantum bit representation is used for antibodies and vaccines where the vaccine population is divided into two sub populations [9]. Genetic operators are used to evolve the first subpopulation and the EDA is applied in the second one to sample the fittest vaccines. The main steps of the proposed algorithm are described in algorithm 2.

The algorithm starts by initializing both the quantum antibody population $Q(t)$ and the quantum vaccine population $V(t)$ followed by cloning and mutataing antibodies to be then decoded for evlauation. Additional steps to the simple QICA, like vaccine decoding and sampling will be described in details. The quantum vaccine population $V(t)$ is initialized in the same way with n quantum vaccines where n is the number of grids that the decision space is divided to and $n = (D_1 * D_2 * \cdots * D_d)$ with d which is the number of dimensions.

Algorithm 2. The proposed Algorithm (QICA-V with EDA)

```
Initialize the quantum antibody and vaccine populations, Q(t) and V(t).
Initialize t=1 as first iteration
while termination condition is not satisfied do
    Apply the clonal and quantum mutation operators over the $Q(t)$ to get $Q'(t)$
    Produce $B'(t)$ by observing $Q'(t)$.
    Decode $V(t)$ to get $V_2$.
    Divide $V_2$ into two subpopulations, $V_2'$ and $V_2''$.
    Select the farthest vaccines from $V_2'$ as the current $V_best$.
    Estimate probability distribution of the $V_best$.
    Sample the distribution to get the $newV_2'$.
    Apply the genetic operators over the $V_2''$ to get the $newV_2''$.
    Build the $newV_2$ by merging the $newV_2'$ and $newV_2''$.
    Apply vaccination over $B(t)$ using the $newV_2$ to get $BV(t)$.
    Apply clonal selection operator over $BV(t)$ to get $Q(t+1)$.
end while
```

- **Initilization**: Quantum antibodies and vaccines populations are created where $V(t)$ is initialized with n quantum vaccines where n is the number of grids that the decision space is divided to. Quantum antibodies $Q(t)$ are cloned and mutated to get $Q'(t)$ using the clonal operator θ where,

$$\theta(Q_t) = [\theta(q_1), \theta(q_2), \ldots, \theta(q_m)] \tag{1}$$

- **Vaccine Selection and vaccination**: Hamming distance is used to compute the distance between the vaccines and antibodies to evaluate the farthest vaccines. Vaccines with higher hamming distances from all antibodies are selected into V_best set. Vaccines in this set are used to apply the injection process over the mutated antibodies clones.
- **Vaccine Sampling**: EDA estimates the probability distribution of the next iterations best vaccines from the current V_best. It uses the mean and standard deviation (sd) of the vaccines in V_best to construct its model.
- **Clonal selection**: The best antibodies from the vaccined antibodies population and selected to form Q(t+1) to proceed to a new iteration.

4 Experiemntal Results

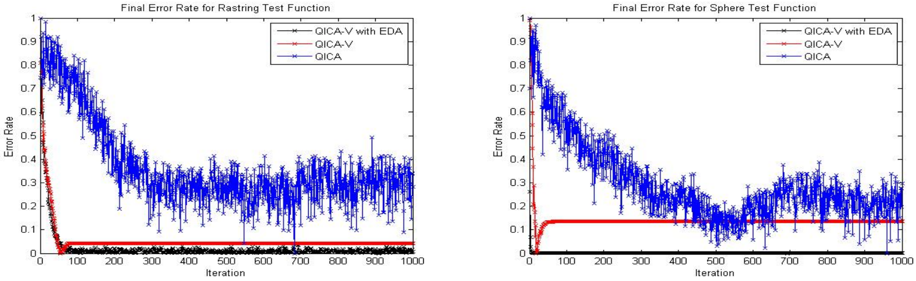
This section introduces the implementation and evaluation of the proposed algorithm. An intial number of the antibodies was set to 5 where 1000 iterations were done. Each antibody has a clone scale of 5 and a probabailty of mutation of 0.5. The proposed algorithm is implemented and evaluated using eight benchmark test problems where the first four are unimodal and last four are multimodal problems [9]. A set of four evaluation indicators are used in this paper where they are computed for the QICA-V with EDA over all the test functions. These indicators include the best (B) and worst solutions (W) found in addition to the average fitness (AF) and average standard deviation (AS) of all solutions. The results are then compared with their found in the literature for the QICA and the QICA-V algorithms as in table 1. Table 1 shows that the QICA-V with EDA performs better than QICA and QICA-V in all experiments except for the unimodal Shwefels and multi modal Shwefel functions. The algorithm is able to achieve the optimal solution of the problems with high degree of diversity represented in the high standard deviation values. The EDA sampling mechanism of the proposed algorithm proved its effectiveness over the vaccine operator of the QICA-V in reaching optimal solutions in multimodal problems and maintaining better performance with reduced complexity.

The error rate of the proposed algorithm and compared algorithms are recorded and visualized for all test problems. Due to limit pages Fig. 1(a)& 1(b) show error rates for some benchmark test problems. The proposed algorithm performance is the best over the other algorithms in achieving optimal solutions at earlier iterations. Although the multimodal property of the Rastring problem, QICA-V with EDA was the best and the quickest to achieve optimality where QICA-V takes more iterations to achieve it. QICA has the worst performance where it failed to achieve the optimality in the multi modal Rastring problem and converges to it in Sphere problem. The population dynamics was also captured and visualized to check how the solutions are evolved through the evolutionary process. The population dynamics of the first 3000 evaluations of the QICA-V, QICA and our algorithm for a sample of multimodal and unimodal test problems are shown in Fig.2, 3 &4.

As shown in Fig.2, 3 &4, QICA-V with EDA was able to converge to optimal solutions through the first evluations for the multi modal problems although the QICA-V failed to do the same. Solutions have high degree of variation due to

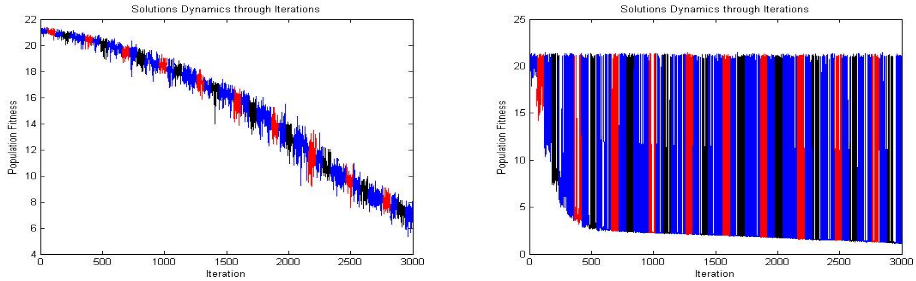
Table 1. Experimental Results of all algorithms

	Measure	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**
QICA-V with EDA	**B**	19.89	2.12	2191.80	0	-5317.31	0	0	0.05
	AF	14.03	1.49	154905.88	59.24	-648.16	129.73	6.13	236.64
	AS	5.77	0.47	193707.88	96.89	749.71	202.43	9.2	390.10
	W	0.8	-0.62	3529932.45	366.15	2844.27	21.60	21.61	1482.22
QICA-V	**B**	19.85	2.07	35.17	0	-12566.88	0.18	0.18	1.37
	AF	12.43	1.69	7293.64	3.14	-12367.85	1.01	0.94	19.90
	AS	4.47	0.5	6160.19	0.5	20.97	0.04	0.04	2.14
	W	0.8	-0.62	1429705.25	391.35	905.60	21.43	21.43	1543.21
QICA	**B**	19.85	2.07	17348.69	44.28	-5353.18	189.66	17.5	279.58
	AF	11.84	1.4	617023.70	176.80	5.19	412.5	20.73	827.96
	AS	5.52	0.59	685738.53	35.27	1172.80	45.07	0.28	120.03
	W	0.8	-0.62	8646771.22	435.9	4845.16	21.50	21.55	1758.54



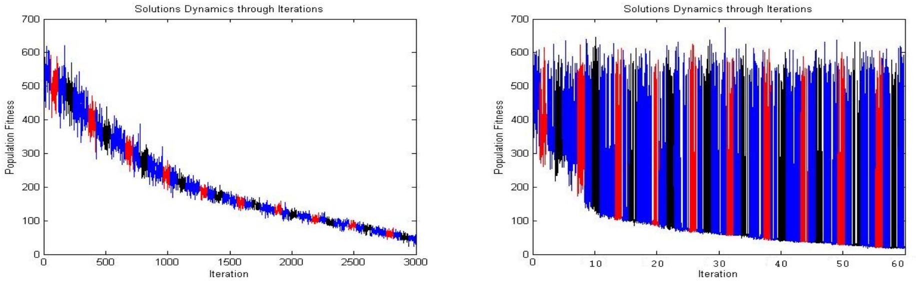
(a) Error Rate for Rastring problem. (b) Error Rate for Sphere problem.

Fig. 1. Error Rate for Rastring & Sphere function



(a) Pop. dynamics using QICA-V. (b) Pop. dynamics using proposed algorithm.

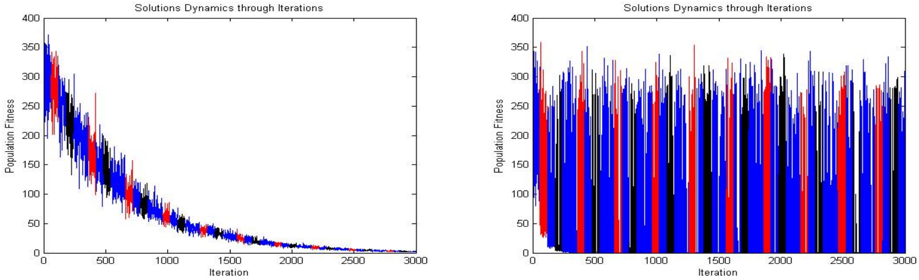
Fig. 2. Population dynamics of Ackely function



(a) Pop. dynamics using QICA-V. (b) Pop. dynamics using proposed algorithm.

Fig. 3. Population dynamics of Rastring function

the EDA sampling mechansim where low vaired solutions obtained by QICA-V using the genetic cross over and mutataion. For unimodal functions, QICA-V has better performance due to the simple problems structure but EDA sampling was again better. For simple two dimension problem, both algorithms behave almost the same either in convergence speed or solutions diversity.



(a) Pop. dynamics using QICA-V. (b) Pop. dynamics using proposed algorithm.

Fig. 4. Population dynamics of Sphere function

5 Conclusions

In this paper, we proposed a new bio inspired algorithm that integrates quantum vaccine immune clonal algorithm with EDA (QICA-V with EDA). It employs immune concepts and the quantum computing principles with the aid of vaccine operator and EDA sampling mechanism. The quantum representation and vaccination helped in improving the search capabilities of the algorithm and the fitness of solutions. The EDA sampling helped in improving the diversity between solutions with reduced complexity and execution time. The performance of the proposed algorithm was analyzed and the results verified that it outperformed QICA and the QICA-V. It was able to produce high quality diversified solutions for both unimodal and multimodal benchmark problems. For further research, extensive experiments with detailed analysis are needed as well as an implementation of real applications.

References

1. Liu, F., Liu, J., Feng, J., Zhou, H.: Estimation Distribution of Algorithm for Fuzzy Clustering Gene Expression Data. In: Jiao, L., Wang, L., Gao, X.-b., Liu, J., Wu, F. (eds.) ICNC 2006. LNCS, vol. 4222, pp. 328–335. Springer, Heidelberg (2006)
2. Yuan, G.L., Xue, Y.G., Liang, Q.J.: The Design of Adaptive Immune Vaccine Algorithm. Journal of Advanced Materials Research, 308–310 (2011)
3. Talbi, H., Batouche, M., Draa, A.: A Quantum-Inspired Evolutionary Algorithm for Multi objective Image Segmentation. World Academy of Science, Engineering and Technology 31, 205–2010 (2007)
4. Sun, J., Zhang, Q., Tsang, E.P.K.: DE/EDA: A New Evolutionary Algorithm for Global Optimization. Information Sciences 169(4), 249–262 (2005)
5. Gao, J., Fang, L., He, G.: A Quantum-Inspired Artificial Immune System for Multiobjective 0-1 Knapsack Problems. In: Zhang, L., Lu, B.-L., Kwok, J. (eds.) ISNN 2010, Part I. LNCS, vol. 6063, pp. 161–168. Springer, Heidelberg (2010)
6. Greensmith, J., Whitbrook, A.M., Aickelin, U.: Artificial Immune Systems. Computing Research Repository (CoRR) 1006, 4949 (2010)

7. YangYang, L., LiCheng, J.: Quantum-Inspired Immune Clonal Algorithm for SAT Problem. Chiness Journal of Computers 2 (2007)
8. Lukac, M., Perkowski, M.: Evolving Quantum Circuits using Genetic Algorithm. In: Proc.of NASA/DOD Workshop on Evolvable Hardware, Washington (2002)
9. Soliman, O.S., Rassem, A.: Quantum Vaccine Immune Clonal Algorithm with EDA Sampling. In: The Proceeding of the Annual Conf. ISSR, Cairo University (2011)
10. Larraiiaga, P., Etxeberria, R., Lozano, J.A., Peiia, J.M.: Combinatorial Optimization by Learning and Simulation of Bayesian Networks. In: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, pp. 343–352 (2000)
11. Niu, Q., Zhou, T., Ma, S.: A Quantum-Inspired Immune Algorithm for Hybrid Flow Shop with Make span Criterion. Journal of Universal Computer Science 15, 765–785 (2009)
12. Yang, S., Wang, M., Jiao, L.: Quantum-inspired immune clone algorithm and multiscale Bandelet based image representation. Journal Pattern Recognition Letters 13, 1894902 (2010)
13. Huil, W., Xiaojun, B., Lijun, Y., Lijun, Z.: An adjustable threshold immune negative selection algorithm based on vaccine theory. Journal of Harbin Engineering University (2011)
14. Woldemariam, K.M., Yen, G.G.: Vaccine-Enhanced Artificial Immune System for Multimodal Function Optimization, Systems, Man, and Cybernetics, Part B: Cybernetics. IEEE Transactions 40, 218–228 (2010)
15. He, X., Zeng, J., Xue, S., Wang, L.: An New Estimation of Distribution Algorithm Based Edge Histogram Model for Flexible Job-Shop Problem. In: Yu, Y., Yu, Z., Zhao, J. (eds.) CSEEE 2011. CCIS, vol. 158, pp. 315–320. Springer, Heidelberg (2011)
16. Li, Y., Liu, F.: A Novel Immune Clonal Algorithm. In: Jiao, L., Wang, L., Gao, X.-b., Liu, J., Wu, F. (eds.) ICNC 2006. LNCS, vol. 4222, pp. 31–40. Springer, Heidelberg (2006)
17. Li, Y., Jiao, L., Gou, S.: Quantum-Inspired Immune Clonal Algorithm for Multiuser Detection in DS-CDMA Systems. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G.-L., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 80–87. Springer, Heidelberg (2006)
18. Ruirui, Z., Jiyin, Z., Tingting, Z., Min, L.: Power Transformer Fault Diagnosis Based on Genetic Support Vector Machine and Gray Artificial Immune Algorithm. In: Proceeding of the CSEE, vol. 31, pp. 56–63 (2011)

Managing Qualitative Preferences with Constraints

Eisa Alanazi[*] and Malek Mouhoub

Department of Computer Science
University of Regina
Regina, Canada
{alanazie,mouhoubm}@cs.uregina.ca

Abstract. Preferences and Constraints co-exist naturally in different domains. Thus, handling both of them is of great interest for many real applications. Preferences usually expressed in qualitative format where a constraint satisfaction problem (CSP) is a well known formalism to handle constraints. In this paper, we investigate the problem of managing both qualitative user preferences and system requirements. We model our preference part as an instance of Conditional Preference networks (CP-nets) and the constraints as CSP. We propose a new method to handle both aspects in an efficient manner. Our method is based on the well-known Arc Consistency (AC) propagation technique. The experiments demonstrate that the new approach can save a substantial amount of time for finding the optimal solution for given preferences and constraints.

1 Introduction

Handling user preferences is crucial step in building successful decision support systems [1,2,3]. Conditional Preference networks (CP-nets) is graphical model to represent conditional qualitative preferences. In addition to preferences, many decisions take place in constrained environment. Therefore, handling both preferences and constraints is of great interest in deploying successful applications. A CSP is a well known framework for representing and solving problems under constraints. Since solving these problems is in general NP-hard, constraint propagation techniques such as Arc Consistency (AC) have been proposed to reduce the size of the search space before and during the search [4,5,6].

Clearly, adding constraints could result in eliminating several scenarios or outcomes from the corresponding CP-net. For example, assume $x_1y_1 \succ x_1y_2 \succ x_2y_1 \succ x_2y_2$ is the pre order for a CP-net involving the values $\{x_1, x_2\}$ and $\{y_1, y_2\}$ for variables X and Y respectively. Now assume that the constraint $C(X, Y)$ does only allow the tuples (x_i, y_j) when $i \neq j$. Obviously, this makes x_1y_1 and x_2y_2 not feasible according to $C(X, Y)$. Therefore, x_1y_1 is no longer the best outcome but x_1y_2 is. These types of inconsistencies can easily be detected and removed

[*] The author is supported by Ministry of Higher Education, Saudia Arabia.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 653–662, 2012.

if the CSP with AC is used to manage hard constraints. More precisely, our approach consists of applying AC first in order to remove some of the inconsistent values. The result is a new CP-net where some domains values are removed from the network. Ideally, this can result in a huge decrease in the search space. By discarding these inconsistent assignments for the CP-net, the discovery of the optimal outcome will be obtained in a shorter period of time. In addition, AC is performed in polynomial time which means that the extra cost due to this propagation technique does not affect the overall running time as demonstrated by the experimental tests we conducted on randomly generated instances and reported in this paper. Note that if one of the variables domain becomes empty during the AC process, the CSP is inconsistent in this case and there is no need to look for an optimal solution since a feasible one does not exist.

Many attempts have been made in order to handle constraints and preferences together [7,3,2,8]. In constrained CP-net [7], the CP-net is first converted into a set of hard constraints. The solution to the new constraint network is then the optimal solution to the CP-net. [8] propose a method to approximate the entire acyclic CP-net to SCSP instance. However, our approach is different in sense that it works in both acyclic and cyclic CP-nets. Also, we are not aware of any attempt to prune variables values and CP statements before searching for best outcome.

The rest of the paper is structures as follows. Literature review on existing work is first covered in the next section. The relation between CP-nets and CSPs is then investigated on section three. Following that, the managing aspect of constraints and preferences is discussed and the new approach is represented. The fifth section illustrates our method by detailed example for the dress-up game. Experimental results of the new technique are shown in the sixth section. Some possible future work and the conclusion is listed in the final section.

2 Background

2.1 Conditional Preferences Networks (CP-Nets)

A Conditional Preferences network (CP-net) [1,9] is a graphical model to represent qualitative preferences statements including conditional preferences such as: "*I prefer A to B when X holds*". A CP-net works by exploiting the notion of preferential independency based on the *ceteris paribus* (with all other things being without change) assumption. Ceteris Paribus (CP) assumption gives us a clear way to interpret the user preferences. For instance, I prefer A more than B means I prefer A more than B if there was no change in the main characterstics of the objects. A CP-net can be represented by a directed graph where nodes represent features (or variables) along with their possible values (variables domains) and arcs represent preference independencies among features. Each variable X is associated with a ceteris paribus table (denoted as $CPT(X)$) expressing the order ranking over different values of X given the set of parents $Pa(X)$. An outcome for a CP-net is an assignment for each variable from its domain. Given a CP-net, the users usually have some queries about the set of preferences represented. One of the main queries is the best outcome given the set of preferences. We say outcome o_i is better than outcome o_j if there is a sequence of worsening flips going from o_i to o_j [7]. A Worsening flip is a change in the variable value to a less preferred value according to the variable's CPT.

2.2 Constraint Satisfaction Problems (CSPs)

A Constraint Satisfaction Problem (CSP) [6] is a well-known framework for constraint problems. More formally, a CSP consists of a set of variables each defined on a set of possible values (variable domain) and a set of relations restricting the values that each variable can take. A solution to a CSP is a complete assignment of values to variables such that all the constraints are satisfied.

2.3 Arc Consistency (AC)

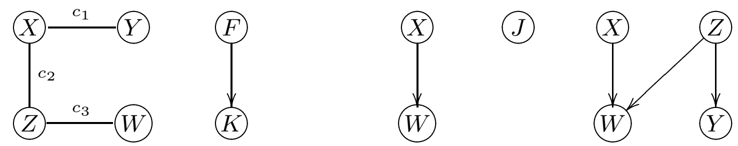
A CSP is known to be an NP-Hard problem. In order to overcome this difficulty in practice, several constraint propagation techniques have been proposed [6,5]. The goal of these techniques is to reduce the size of the search space before and during the search for the solution to the CSP. One of the well-known constraint propagation techniques is called Arc Consistency (AC) [4]. The aim of AC is to enforce a 2 consistency over the constraint problem. More precisely, the 2 consistency consists in making sure that for each pair of variables (X, Y) sharing a constraint, every value a from X's domain has a corresponding value in Y's domain such that the constraint between X and Y is satisfied, otherwise a is eliminated.

3 The Relation between CSPs and CP-Nets

Preferences and constraints represent different but closely related types of information. Preferences can be viewed as desires or wishes where constraints are strict requirements. Both of them are closely linked in a sense that some preferences can be promoted to constraints and the vise versa. Another view is to see preferences as tolerant constraints [2]. When given a particular CP-net N and a set of constraints C, the relation between N and C can fall into one of the following three cases:

1. N does not exist in C. Here, N has no common variables or attributes with C. Therefore, N can be solved via different typical CP-net algorithms.
2. N partially in C. In this case, N has some features or variables that exist in C.
3. N fully exists in C. When N fully exists in C, all attributes in$N$ are also in C.

In the last two cases, there are always subset variables V which exist both in N and C. These types of relation are of interest in this paper. Henceforth, for every CP-net and CSP there are some variables that are shared between them. When considering CSP in Figure 1a, one possible representation for each case is shown in Figures 1b 1c and 1d.



(a) Simple CSP **(b)** No Exitence **(c)** Partially **(d)** Fully

Fig. 1. The Relation Between CP-net and CSP

4 Managing Preferences and Constraints

4.1 CP-Net under Constraints

While a CP-net is a powerful model for representing qualitative preferences [2], managing both hard constraints and preferences is required in many real world applications [11,10]. In these situations, it is important to determine the best solution to the CP-net with respect to the set of hard constraints that we represent with a CSP. It should be noted that satisfying a set of hard constraints is often more important than satisfying the user's preferences statements due to the nature of the preferences and constraints. Constraints mostly represent strict system requirements while preferences represent a pre-order likelihood over a set of features. As a result, in our proposed approach, with respect to any CP-net there is a CSP behind it representing the hard constraints. Following this representation, when looking for the best outcome we will first run arc consistency in order to remove some inconsistencies (which will reduce the size of the search space) and then look for the best outcome in the simplified CP-net.

4.2 Arc Consistency for CP-Nets

Many instantiations can lead to inconsistent assignments in the presence of constraints in addition to a CP- net. Therefore, detecting the inconsistent assignments in a CP-net from the beginning should lead to a reduction in the complexity of the problem. The CSP framework comes with a large number of techniques to detect inconsistencies in the domain. However, a CP-net does not offer a systematic way to detect the set of preferences that are inconsistent with the constraints. Determining the set of inconsistent preference statements will eventually lead to finding the solution faster or declaring the problem to be inconsistent. Thus, adapting these techniques to a CP-net seems to be a useful technique. As result, a new method is defined to handle constraints over a CP-net based on the Arc Consistency (AC) technique.

We say CP-net N with CSP C is *arc consistent CP-net* (ACCPnet) if for every domain value x for variable $X \in Vars(N)$ x is either satisfied by C or X does not exist in C. Though we consider the query of finding the best outcome for a particular CP-net, our method works *regardless* of the query posed by the user. Our main contribution is reducing the search space needed for reasoning about CP-nets in the presence of constraints without posing any restriction to the type of query. ACCPnet is the result of updating AC changes to CP-net. In order for a given CP-net to reflect the AC changes we propose an algorithm (Algorithm 1) that traverse over the original CP-net and CSP and return ACCPnet instance of the problem.

Checking the consistency of a variable X is straightforward. We simply check the cardinality of the domains, i.e. $|dom(X)| \neq |dom(\mathbf{X})|$. Since for each CP-net variable X there is a set of parents $Pa(X)$, we refer to the set of different instantiations of the parents in CPT(X) as $pa(X)$. The time complexity of the this approach is as follows assuming N is the number of variables, e the number of constraints, d the largest domain size of the variables and m the largest CPT in the network. We use AC-3 [12] for the arc consistency and the corresponding cost is $O(ed^2)$. In the worst case scenario, the cost of the ACCPnet algorithm is $N(d+m)$.

Algorithm 1. ACCPnet(CPN, CSP)

```
Let $\mathbf{V}_{csp}$ be CSP variables
  Let $\mathbf{V}_{cpn}$ be CPN variables
  Let $\mathbf{V}_{shared} = \mathbf{V}_{csp} \cap \mathbf{V}_{cpn}$
  for each $X \in \mathbf{V}_{shared}$
      if $\neg isConsistent(X, \mathbf{X})$
         for each $x_i \in dom(X)$
            if $x_i \notin dom(\mathbf{X})$
              { remove $x_i$ from $dom(X)$ and $CPT(X)$
                for each $Y \in children(X)$
                for each $S \in CPT(Y)$
                  if $x_i \in pa(S)$
                  remove $S$
              }
     return CPN
```

4.3 Finding the Optimal Outcome

To show the effectiveness of the ACCPnet, we consider the problem of finding best outcome given set of constraints. The optimal solution for an acyclic CP-net N is the one with the minimum worsening flips according to N [1,9]. The best outcome can be computed by assigning each variable to its most preferred value throughout the network . In the presence of constraints, an optimal assignment A can be infeasible. In this case, a solution should be investigated where A satisfies the set of constraints C while minimizing the number of worsening flips.

5 Illustrative Example: The Dress-Up Game

In this section we illustrate our proposed approach through the dress up game which provides the user with sets of clothes, accessories and shoes. The user will then use his free style Mix and Match imagination to create a complete outfit. In order to assist the user to have an appropriate outfit and be in a fashion trend, we can enhance the dress up game by adding rules, tips and advices from fashion designers. This latter information can be modeled as constraints and preferences respectively through a CSP and a CP-net as we will show in the next two subsections.

5.1 CSP Representation

Figure 2 shows the constraint graph for a given dress up game problem. Figures 3 and 4 list the domains of the variables and the constraints for this problem.

5.2 CP-Net Representation

Let us consider the following preferences for our dress up game.

- *We always prefer to wear "casual shoes" and "boots" instead of "sandals", "runners" and "pumps".*
- *We like handbags "HB3" and "HB4" the most*
- *For matching clothes (Top & Bottom constraints), we like "Blouses with Skirts" and "t-shirts with capris" and "jackets with jeans" the most.*
- *As for the Set & Shoes constraint, we prefer a "skirt outfit with boots" and a "pant-suit with casual wear" for the rest.*

Figure 5 shows the CPT tables corresponding to these preferences. The relation $\succeq$ between two pairs of values states that they are equally preferred to the user.

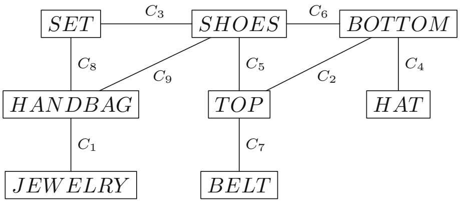


Fig. 2. CSP for the Dress-up Example

SHOES	{sandal=1, pump=2, boot=3, running=4, casual=5}
SET	{skirt_suit=6, dress=7, pant_suit=8}
TOP	{blouse=9, shirt=10, t-shirt=11, tank=12, jacket=13}
BOTTOM	{pant=14, jean=15, skirt=16, capri=17, short=18}
HAT	{bucket=19, visor=20, toyo=21, baseball=22}
BELT	{B1=23, B2=24, B3=25, B4=26}
JEWELRY	{J1=27, J2=28, J3=29, J4=30}
HANDBAG	{HB1=31, HB2=32, HB3=33, HB4=34}

Fig. 3. Domain Values for the Dress-up Example

JEWELRY	HANDBAG
27	32
28	32
29	33
29	34
29	31
30	31
30	32
30	33

BOTTOM	SHOES
14	5
14	2
15	4
16	3
16	2
17	1
18	1
18	4

TOP	BELT
13	26
12	25
12	26
11	25
9	24
9	25
10	24
10	26

TOP	BOTTOM
9	14
9	16
10	14
10	16
11	15
11	17
11	18
12	18
12	17
13	15

HAND_BAG	SET
32	7
31	6
34	8
33	7

HANDBAG	SHOES
31	1
32	2
33	2
34	5
33	3
34	1

HAT	BOTTOM
19	16
20	17
20	18
22	18
22	15
21	16

SET	SHOES
6	2
6	3
7	2
8	2
8	5

TOP	SHOES
9	2
10	5
11	1
11	4
12	1

Fig. 4. Dress-up Constraints

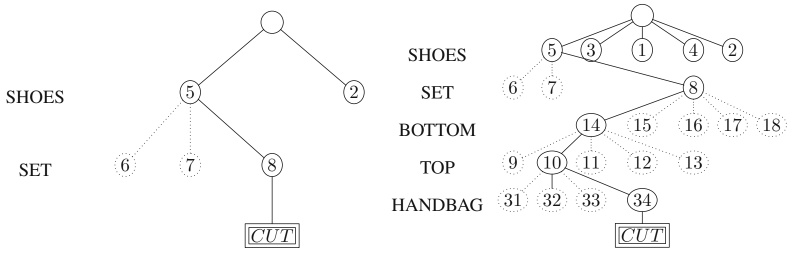
Attribute	Preference Table
SHOES	$5 \succeq 3 \succ 1 \succ 4 \succ 2$
BOTTOM	$15 \succeq 16 \succeq 16 \succ 17 \succ 14 \succ 18$
SET	$5 : 6 \succ 7 \succ 8$ $3 : 8 \succ 7 \succ 6$ $otherwise : 7 \succ 6 \succ 8$
TOP	$15 : 13 \succ 12 \succ 11 \succ 10 \succ 9$ $17 : 11 \succ 9 \succ 10 \succ 12 \succ 13$ $otherwise : 9 \succ 10 \succ 11 \succ 12 \succ 13$
HB	$33 \succeq 34 \succ 32 \succ 31$

Fig. 5. The set of Conditional Preference Tables (CPTs)

5.3 Determining the Best Outcome

Let us assume the following topological ordering for the CP-net as {SHOES,SET,BOTTOM,TOP,HANDBAG}. In order to demonstrate the importance of arc consistency, the problem of finding the optimal outcome is shown in the following subsections with and without this local consistency technique.

Finding the Best Outcome without AC. Figure 6b shows the search space where the dotted arcs and nodes represent the inconsistent values for the corresponding variables. The most preferred values for SHOES, which is SHOES = 5, is first taken. An attempt is then made to extend this assignment to the most preferred value SET where SET = 6 and SET = 7 is inconsistent with SHOES = 5. Hence, we have SHOES = 5 and SET = 8 as a partial assignment with worsening flips = 2. This partial assignment is extended to BOTTOM as follows: {5,8,14}. After considering the other consistent possibilities, this partial assignment is extended to the following complete assignment which is inconsistent according to the CSP: {5, 8, 14, 10, 34}. The next value to consider is



(a) After **(b)** Before

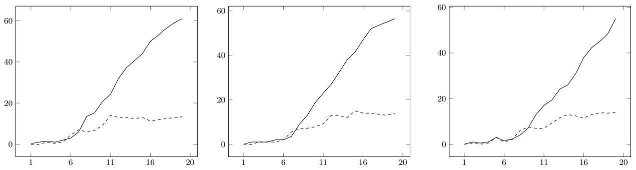
Fig. 6. CP-net before and after arc consistency

3, but it is found to be inconsistent, along with 4 and 1. Finally, SHOES = 2 works with worsening flips = 3. The complete assignment {2,7,9,16,33} is then obtained and is actually consistent with the CSP. The set of optimal outcomes is any solution where $SHOES = 2$, $SET = 7$, $BOTTOM = 16$, $TOP = 9$ and $HANDBAG = 33$ with 3 worsening flips.

Using AC to Find the Best Outcome. Figure 6a shows the same search space for the CP-net after applying AC. The value 14 is removed from the domain of BOTTOM which will prevent us from considering the partial assignment {5, 8, 14} as we did in the previous subsection. As we can easily see from the figure, the search space is reduced and the inconsistency is detected here in advance.

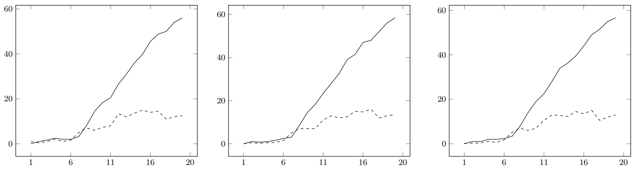
6 Experimentation

We evaluate the performance of our proposed technique on different randomly generated problems. We have implemented a solver to handle both CP-net and CSP instances. Our solver is coded in Java under Netbeans IDE. The operating system specifications are Mac OS X versions 10.6.7 with 2GHz Intel i7 processor and RAM with 4GB. We have conducted 9 experiments to investigate both the optimal solution and search space (number of possibilities) problems. Eliminating some domain variables will indeed result in reducing the number of possibilities. The goal of the space experiments is to show how our method can drastically reduce the space needed for given CP-net regardless of the imposed query (i.e. finding best outcome). Note that some problems are inconsistent and thus we neglect them from our experiment calculations. This means there is no solution which satisfy hard constraints. We generate 100 problems for each experiment using Model RB [13]. The reason for choosing this model is that it has exact phase transition and the ability to generate asymptotically hard instances. We focus on two important parameters: tightness and density. The constraint tightness is defined as the ratio of the number of allowed tuples to the total number of possible combinations (cartesian product) [14]. In each experiment



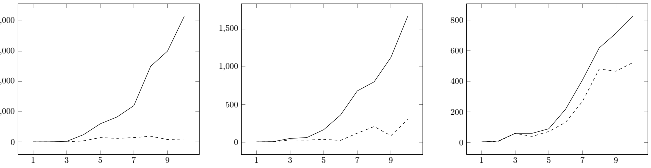
(a) density75 **(b)** desnity50 **(c)** desnity25

Fig. 7. Experiments with different *Density* and fixed *Tightness*



(a) tightness25 **(b)** tightness50 **(c)** tightness75

Fig. 8. Experiments with different *Tightness* and fixed *Desnity*



(a) possibilities25 **(b)** possibilities50 **(c)** possibilities75

Fig. 9. Possibilities to Different *Tightness* with fixed *Desnity*

we alter one of the parameters and fix the other. For our computational limit, we consider problems with 20 variables in each network. However the same conclusion can be reached with larger problems. Figure 8 shows the average for finding optimal solution for CP-net in with different density parameters. Likely Figure 7 shows three experiments when we have tightness equal to 25%, 50% and 75%. The x-axis represents number of variables where the y-axis represents the execution time in milliseconds. Figure 9 represents the number of possibilities when varying the tightness and setting the density to 50%.

7 Conclusion and Future Work

In this paper, a discussion was initiated regarding the relation between the CSP and the CP-net. Then, the consistency aspect of the CP-net was introduced along with a method to apply arc consistency to the CP-net. The ACCPnet algorithm with which to update the CP-net to reflect the new domains in the CSP was introduced and discussed. It was shown that applying arc consistency to the CP-net can reduce the search space and the time needed for finding the best outcome. This work presents the believe that the usage and the efficiency of CP-net can be improved by considering different CSP techniques. Future work includes handling CP-net with quantitative preferences, generalizing the method proposed to handle soft constraints and managing CP-net in the presence of dynamic and conditional constraints.

References

1. Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., Poole, D.: Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. J. Artif. Intell. Res. 21, 135–191 (2004)
2. Rossi, F., Venable, K.B., Walsh, T.: Preferences in constraint satisfaction and optimization. AI Magazine 29, 58–68 (2008)
3. Boutilier, C., Brafman, R.I., Hoos, H.H., Poole, D.: Preference-based constrained optimization with cp-nets. Computational Intelligence 20, 137–157 (2001)
4. Mackworth, A.K.: Consistency in networks of relations. Artificial Intelligence 8, 99–118 (1977)
5. Kumar, V.: Algorithms for constraint satisfaction problems: A survey. AI Magazine 13, 32–44 (1992)
6. Dechter, R.: Constraint processing. Elsevier Morgan Kaufmann (2003)
7. Prestwich, S., Rossi, F., Venable, K.B., Walsh, T.: Constrained cpnets. In: Faltings, B.V., Petcu, A., Fages, F., Rossi, F. (eds.) CSCLP 2004. LNCS (LNAI), vol. 3419, Springer, Heidelberg (2005)
8. Domshlak, C., Rossi, F., Venable, K.B., Walsh, T.: Reasoning about soft constraints and conditional preferences: complexity results and approximation techniques. CoRR abs/0905 (2009)
9. Brafman, R.I., Dimopoulos, Y.: A new look at the semantics and optimization methods of cp-networks. In: IJCAI, pp. 1033–1038 (2003)
10. Mouhoub, M., Sukpan, A.: Managing temporal constraints with preferences. Spatial Cognition & Computation 8, 131–149 (2008)
11. Alanazi, E., Mouhoub, M.: Arc consistency for cp-nets under constraints. In: FLAIRS Conference (2012)
12. Bessière, C., Régin, J.C., Yap, R.H.C., Zhang, Y.: An optimal coarse-grained arc consistency algorithm. Artif. Intell. 165, 165–185 (2005)
13. Xu, K., Li, W.: Exact phase transitions in random constraint satisfaction problems. Journal of Artificial Intelligence Research 12, 93–103 (2000)
14. Beek, P.V., Dechter, R.: Constraint tightness and looseness versus local and global consistency. Journal of the ACM 44 (1997)

A Bio Inspired Fuzzy K-Modes Clustring Algorithm

Omar S. Soliman, Doaa A. Saleh, and Samaa Rashwan

Faculty of Computers and Information, Cairo University, Egypt
dr.omar.soliman@gmail.com

Abstract. This paper proposes a bio inspired fuzzy K-Modes clustering algorithm using fuzzy particle swarm optimization (FPSO) and fuzzy k-modes (FK-Modes) algorithm for clustering categorical data. It integrates concepts of FK-Modes algorithm to handle the uncertainty phenomena and FPSO to reach global optimal solution of clustering optimization problem. The proposed FPSO-FK-Modes algorithm was implemented and evaluated using slandered benchmark data sets and performance measures. Experimental results showed that the proposed FPSO-FK-Modes algorithm performed well compared with FK-modes and Genetic FK-modes (GA- FK-modes) algorithm using adjusted rand index.

Keywords: Fuzzy clustering, Categorical data, FK-modes, FPSO, ARI.

1 Introduction

Data clustering divide objects of a data set into conceptually meaningful groups (clusters), with the objects in a group being similar to one another but very dissimilar to the objects in other groups. Each object has to belong to only one cluster, but, in case of data set with information ambiguity, may be causing to one object can belong to more than one cluster by a degree of membership function (is called fuzzy data clustering). There are numerous algorithms available for doing data clustering. these algorithms can be categorized in various ways such as: hierarchical or partition, deterministic or probabilistic, hard or fuzzy [1], [8]. In the hard clustering algorithms each object is assigned to only one cluster, where in fuzzy clustering, data object is assigned to multiple clusters. The degree of membership function in the fuzzy clusters relies on the closeness of the data object to the center of cluster. The K-modes clustering algorithm is based on K-means paradigm but removes the numeric data limitation whilst preserving its efficiency [11], [17] and [18]. Fuzzy K-modes clustering is an effective algorithm, but the randomization in selecting the center points of cluster causes the falling in a local optimal solution easily. A few algorithms have been proposed in recent years for clustering categorical data. Some of computational intelligent algorithms have been used for improving the clustering performance by finding a global optimal solution for a clustering optimization problem as a Genetic Algorithm (GA) and a Tabu search technique with purpose of improving fuzzy k-modes algorithm [4], [12]. Particle Swarm Optimization (PSO) algorithm is a

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 663–669, 2012.

stochastic global optimization technique [9], [13]. There are some studies that focused on developing hybridized data clustering algorithms by integrating more than one computational such as [6], [4], and [14]. The aim of this paper is to develop a bio inspired fuzzy K-Modes clustering algorithm using fuzzy particle swarm optimization (FPSO) and fuzzy k-modes (FK-Modes) algorithm for clustering categorical data as well as to overcome FK-Modes of categorical data by finding the global optimal solution of the clustering optimization problem. The rest of this paper is organized as follow: Section 2 introduce problem background and related works of FK-modes algorithm and fuzzy PSO. Where, section 3 presents details of proposed algorithm. Section 4 introduces experimental results, measure performance and discussions. Finally, section 5 is devoted to conclusions.

2 Background and Related Works

2.1 Fuzzy K-Mode Algorithm

The fuzzy k-Modes clustering algorithm was introduced in [16]. Unfortunately, the algorithm may be fail in a local optimal solution. Therefore, K-modes could be combined with any one of artificial intelligent techniques to overcome trapping in local optimal solutions. Mathematically, a fuzzy clustering problem can be represented as an optimization problem as follow:

$$Min_{\mu,Z} \quad F(\mu, Z) = \sum_{j=1}^{k} \sum_{i=1}^{n} \mu_{ij}^{\propto} \, d(z_j, x_i) \tag{1}$$

Subject to

$$0 \leq \mu_{ij} \leq 1, \qquad 1 \leq j \leq k, \; 1 \leq i \leq n, \tag{2}$$

$$\sum_{j=1}^{k} \mu_{ij} = 1, \qquad 1 \leq i \leq n, \tag{3}$$

$$0 < \sum_{i=1}^{n} \mu_{ij} < n, \qquad 1 \leq j \leq k, \tag{4}$$

Let D = {x_1, x_2, …, x_n} be a categorical data set with n objects each of which is described by d categorical attributes A_1, A_2, …., A_d. Attribute A_l (1•l • d) has n_j categories, i.e., DOM $(A_l) = \{a_{l1}, a_{l2}, \ldots, a_{lnl}\}$. And let the cluster center be represented by $z_j = \{z_{j1}, z_{j2}, \ldots, z_{jd}\}$ $for\ 1 \leq j \leq k$, where k is the number of clusters.

The simple matching distance measure between x and y in D is defined as:

$$d_C\,(x, y) = \sum_{l=1}^{d} \delta(x_l, y_l) \tag{5}$$

Where x_j and y_j are the j^{th} components of x and y, respectively, and

$$\delta(x_j, y_j) = \begin{cases} 0 & if\ x_j = y_j \\ 1 & if\ otherwise \end{cases} \tag{6}$$

And clusters centroid is updated as:

$$z_{jl} = a_{lr} \in DOM\ (A_l) \tag{7}$$

Where the fuzzy membership matrix is updated as:

$$\mu_{ij} = \frac{1}{\sum_{h=1}^{k}\left[\frac{d(x_i,z_j)}{d(x_i,z_h)}\right]^{\frac{1}{\alpha-1}}} \tag{8}$$

2.2 Fuzzy Particle Swarm Optimization

In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles [2]. [5], [9] and [13]. For a swarm of n particles the ith particle is represented by a position denoted as $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ where n is the number of particles. Except the position, each particle of a swarm is represented in D dimensional space with a velocity $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$. The particles explore in the search space with a velocity that is dynamically adjusted according to its own and neighbor's performances. The standard PSO method updates the velocity and position of each particle according to the equations given below [15].

$$V\,(t+1) = \omega \times V(t) + c_1\, r_1 \times (\text{pbest}(t) - X(t))\ c_2\, r_2 \times (\text{gbest}(t) - X(t)) \tag{9}$$

$$(Xt + 1) = X(t) \oplus V(t+1) \tag{10}$$

In FPSO algorithm, the position of particle, shows the fuzzy relation from set of data objects, $o_i = (o_1, o_2, \dots, o_n)$, to set of cluster centers, $Z_i = (z_1, z_2, \dots, z_k)$. The position matrix $X_{n\times k}$ is defined as follows [5]:

$$X = \begin{bmatrix} \mu_{11} & \cdots & \mu_{1k} \\ \vdots & \ddots & \vdots \\ \mu_{n1} & \cdots & \mu_{nk} \end{bmatrix} \tag{11}$$

In which μ_{ij} is the membership function of the i^{th} object with the j^{th} . And $\mu_{ij} \in [0,1]$ $\forall\, i = 1, 2, \dots.., n;$ $\forall\, j = 1, 2, \dots.., k$ is updated and normalized using eq. 10, and the position matrix $X_{n\times k}$ is updated using eq. 11.

$$X_{normal} = \begin{bmatrix} \frac{\mu_{11}}{\sum_{j=1}^{c}\mu_{1j}} & \cdots & \frac{\mu_{1c}}{\sum_{j=1}^{c}\mu_{1j}} \\ \vdots & \ddots & \vdots \\ \frac{\mu_{n1}}{\sum_{j=1}^{c}\mu_{nj}} & \cdots & \frac{\mu_{nc}}{\sum_{j=1}^{c}\mu_{nj}} \end{bmatrix} \tag{12}$$

3 Proposed Algorithm

The proposed bio inspired algorithm integrates FK-Modes and FPSO. It is divided into several steps starting by initialization required parameters; followed by calculating the position centroids matrix for each particle. Then distance and membership matrix are updated. Followed by, calculating the fitness value for each particle. The next step is to check the stopping criteria either maximum number of iterations or no improvement in gbest. The proposed FPSO-Fk-Modes algorithm is described in algorithm 1.

Algorithm 1. Proposed Bio Inspired algorithm (FPSO-Fk-Modes)

```
Initialize the parameters including population size P, $c_1$, $c_2$, w, $r_1$, $r_2$ and the maximum iterative count.
Create a swarm with P particles (X, pbest, gbest and V are $(n \times k)$ matrices).
Initialize X, V, pbest for each particle and gbest for the swarm.
Select $\alpha$ ($\alpha > 1$); initialize $Z_0$ and the membership function values $\mu_0$, where $\mu_{ij}$, $i = 1, 2, \ldots, n$;  $j = 1, 2, \ldots, k$.
**For** t = 0  to q **do**
Calculate $Z_{t+1}$ for each particle.
Calculate the fitness value of each particle.
  **If**  $F(X_t, Z_{t+1}) = F(X_t, Z_t)$  **then**
  **Stop;**
  **Else;**
Calculate **pbest** for each particle.
Calculate **gbest** for the swarm.
Update the velocity matrix for each particle.
Update ($X_{t+1}$) the position matrix for each particle.
If $(X_{t+1}, Z_{t+1}) = (X_t, Z_{t+1})$ then
Stop
Else
$X_t \leftarrow X_{t+1};$
End if
End if
End for
```

4 Experimental Results

The proposed algorithm is implemented and evaluated using four benchmark datasets (Soybean, Congress, Hays-Rose, and Spect-Heart) of the UCI Machine Learning Repository [3]. It is developed using VC++ and its performance is measured using ARI.

4.1 Results and Discussions

To analyze the performance of the proposed algorithm, the worst, best, and average fitness, and standard deviation are recorded at each run. The average results of 100 independent runs are reported in table 1 and visualized in Fig. 1 for each dataset.

Table 1. Average fitness of the 100 independent runs for each algorithm

Datasets	Cases	FK-modes	GA-FK-modes	FPSO-FK-modes
Soybean	Worst	482.0182	370.7202957	355.1812778
	Best	227.366	204.38398	198.00174
	Average	354.001	287.5521379	276.5915089
	S.D	61.72855	13.08346201	9.033094281
Congress	Worst	4980.828	4317.8952	3743.671772
	Best	2161.184	1514.8589	1442.25635
	Average	3039.44	2916.377	2592.964061
	S.D	38.9121	9.901703242	9.751038615
Hays-Rose	Worst	511.7819	388.99024	387.3447192
	Best	352.2389	371.94218	350.6261747
	Average	404.4479	406.03658	368.9856298
	S.D	36.67275	8.768535992	6.135393869
Spect-Heart	Worst	484.027	378.15338	376.8675984
	Best	316.282	308.12787	301.835744
	Average	363.13	343.15129	338.8516712
	S.D	33.81516	15.71111925	12.46714599

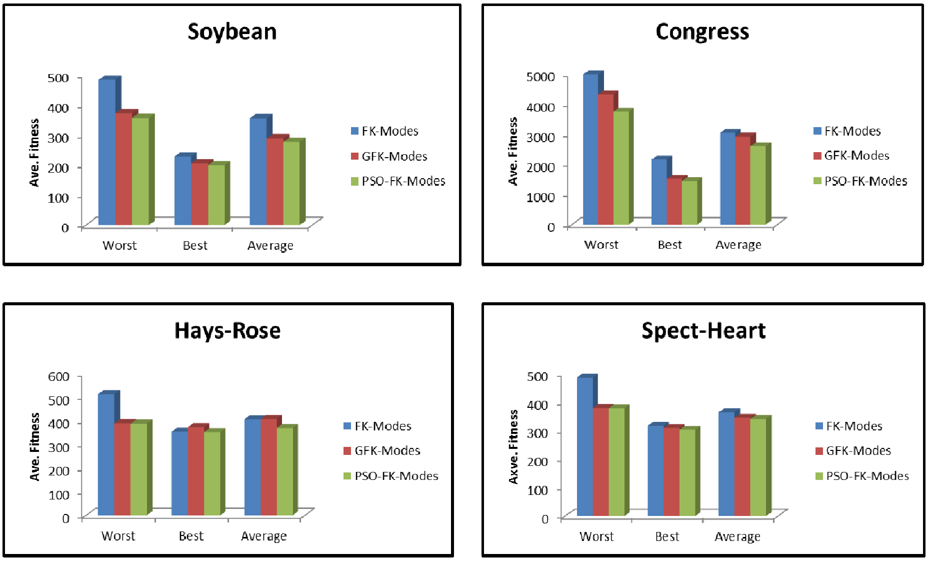


Fig. 1. Comparison results for FK-modes, GA-FK-modes and FPSO- FK-modes algorithms

Fig.1 shows visualized comparison results among FK-modes, GA-FK-modes and FPSO- FK-modes algorithms. Fig. 2(a) shows obtained number of objects which belong to one cluster using three algorithms. As reported in table 1 and, shown in Fig 1 and 2(a) the proposed algorithm is performed well for all datasets.

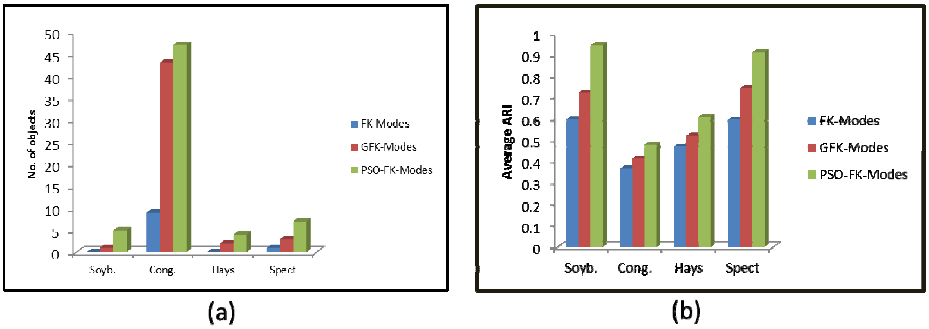


Fig. 2. (a) No. of objects belong to one cluster. (b) Average ARI of each algorithm.

As reported in table 1 and, shown in Fig 1 and Fig. 2(a) the performance of the proposed algorithm is always better among compared two others algorithms. The last evaluation measure of the proposed algorithm is ARI performance measure.

4.2 Performance Measures

The performance of the proposed algorithm was evaluated using the Adjusted Rand Index (ARI) [7], [10].

$$\text{ARI}\ (\gamma) = \frac{\sum_{i,j}^{i\neq j}\binom{n_{ij}}{2} \quad - \quad \frac{\left[\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\right]}{\binom{N}{2}}}{\frac{1}{2}\left[\sum_i\binom{a_i}{2}+\sum_j\binom{b_j}{2}\right]-\frac{\left[\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\right]}{\binom{N}{2}}} \tag{13}$$

The average ARI of the three algorithms for four data sets over 100 independent run is averaged and reported in table 2 and visualized in Fig. 2(b). As reported in table 2 and shown in Fig. 2(b), the proposed FPSO FK-modes algorithm has the highest ARI values among compared algorithms.

Table 2. Average ARI values of the proposed FPSO- FK-modes algorithms for four datasets

Datasets	FK-modes	GA-FK-modes	FPSO-FK-modes
Soybean	0.597659	0.721202471	0.94336228
Congress	0.366902	0.41426397	0.478213
Hays-Rose	0.471414	0.52152586	0.60975001
Spect-Heart	0.596863	0.743266921	0.910440872

5 Conclusions and Future Work

This paper proposed a bio inspired FPSO-FK-modes algorithm that integrates FK-Modes and FPSO for improving the performance of FK-Modes for clustering categorical data and avoiding trapping in a local optimal solution of the optimization clustering problem. It is also aimed to maximize similarity among objects inside the same cluster. The proposed algorithm had been tested and evaluated on four benchmark datasets (Soybean and Congress Voting Hays-Rose, and Spect-Heart) from UCI Repository Machine Learning Datasets. The experimental results showed that the proposed FPSO-FK-modes algorithm performed well compared to FK-modes and GA-FK-modes algorithms. For future work we intend to introduce more experiments and more analysis as well as introducing more performance measures to validate the proposed algorithm.

References

1. Chaturvedi, A., Green, P.E., Carroll, J.D.: K-modes clustering. Journal of Classification (18), 35–55 (2001)
2. Chen, A.L., Yang, G.K., Wu, Z.M.: Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. Journal of Zhejiang University Science 7(4), 604–614 (2006)
3. Blake, C., Merz, C.: UCI Repository Machine Learning Datasets (1998)
4. Gan, G. Wu, J., Yang, Z.: A genetic fuzzy k-Modes algorithm for clustering categorical data. Journal of Expert Systems with Applications (36), 1615–1620 (2009)
5. Hesam, I., Ajith, A.: Fuzzy C-means and Fuzzy Swarm for Fuzzy Clustering Problem. Journal of Expert Systems with Applications (38), 1835–1838 (2011)
6. Michael, K.L., Liping, J.: A new Fuzzy K-Modes Clustering Algorithm for Categorical Data. Int. Journal of Granular Computing, Rough Sets and Intelligent Systems (1), 105–119 (2009)
7. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification, 193–218 (1985)
8. Kweku, M., Osei, B.: Towards supporting expert evaluation of clustering results using a data mining process model. Journal of Information Sciences 180, 414–431 (2010)
9. Bajpai, P., Singh, S.N.: Fuzzy Adaptive Particle Swarm Optimization for Bidding Strategy in Uniform Price Spot Market. IEEE Transactions on Power Systems 22(4) (2007)
10. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66, 846–850 (1971)
11. Aranganayagi, S., Thangavel, K.: Extended K-Modes with Probability Measure. International Journal of Computer Theory and Engineering 2(3), 1793–8201 (2010)
12. Khan, S.S., Kant, S.: Computation of Initial Modes for K-modes Clustering Algorithm using Evidence Accumulation. In: IJCAI 2007, pp. 2784–2789 (2007)
13. Shi, Y., Eberhart, R.C.: Parameter Selection in Particle Swarm Optimization. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 591–600. Springer, Heidelberg (1998)
14. Wenting, C., Hai, Z., Fengling, D.: Cluster Analysis Based on Fuzzy K-Modes and Immune Genetic Algorithm. Journal of Computer Technology and Developmen 19(2), 151–153 (2009)
15. Shi, Y., Eberhart, R.C.: Fuzzy Adaptive Particle Swarm Optimization. In: Proceedings of the. IEEE Congress on Evolutionary Computer, Seoul, Korea (2001)
16. Huang, Z., Ng, M.K.: A fuzzy k-modes algorithm for clustering categorical data. IEEE Transaction on Fuzzy Systems 7(4), 446–452 (1999)
17. Huang, Z.: A fast clustering algorithm to cluster very large categorical data sets in data mining. In: Proc. SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, pp. 1–8 (1997)
18. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery 2(3), 283–304 (1998)

Evaluating SPAN Incremental Learning for Handwritten Digit Recognition

Ammar Mohemmed[1], Guoyu Lu[3], and Nikola Kasabov[1,2]

[1] Knowledge Engineering Discovery Research Institute,
Auckland University of Technology, New Zealand
{amohemme,nkasabov}@aut.ac.nz
http://www.kedri.info
[2] Institute for Neuroinformatics, ETH and University of Zurich
[3] Department of Information Engineering and Computer Science
University of Trento, Italy

Abstract. In a previous work [12, 11], the authors proposed SPAN: a learning algorithm based on temporal coding for Spiking Neural Network (SNN). The algorithm trains a neuron to associate target spike patterns to input spatio-temporal spike patterns. In this paper we present the details of experiment to evaluate the feasibility of SPAN learning on a real-world dataset: classifying images of handwritten digits. As spike encoding is an important issue in using SNN for practical applications, we discuss few methods for image conversion to spike patterns. The experiment yields encouraging results to consider the SPAN learning for practical temporal pattern recognition applications.

Keywords: Spiking Neural Networks, Supervised Learning, Nuerocomputing, Spatiotemporal pattern recognition.

1 Introduction

Driven by the emerging need for systems that can behave autonomously and adaptively through learning, research is turning to biological intelligence for better solutions. The knowledge about how the brain is functioning that becomes available due to the discoveries of neuroscience is inspiring researchers to mimic the brain, at different levels, to create more efficient methods and systems. Spiking Neural Networks (SNN) [5, 9], considered the third generation of artificial neural networks, is an important tool to model many functional aspects of the brain. Furthermore, SNN models have been investigated for a number of computer applications including computer vision [18, 3], speech recognition [17], autonomous robots [4, 15] and others.

Most of real-world data is represented as static or dynamic continuous values. In the latter case, data changes in time and space where useful knowledge can be extracted only after a certain time period of observing the data. An example is extracting information from video data, such as human action recognition for human-robotic interaction or security/health surveillance.

SNN internally use spikes to communicate, where information is encoded in the time of the spikes. That makes SNN to be suitable for spatio-temporal data

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 670–677, 2012.

processing. However, for SNN to be applicable for real-world problems, input data needs to be transformed into spikes before it can be processed in SNN.This conversion should be done properly such that the inter-class/intra-class relationships between data categories are preserved, otherwise the recognition task using SNN will be hard to yield accurate results.

In fact how to encode information into spikes is a challanging problem that extends to a deep research field in neuroscience. According to previous studies such as in [6] and recent one [14], temporal coding whereby information is encoded into precise time of the spikes, plays a significant role in the neural code of the brain especially in the visual system.

SPAN [12, 11], is a learning algorithm for spiking neural networks which is based on encoding input information as precise time of spikes (Temporal coding). This is opposite to rate coding where information is coded in the mean firing rate of the neurons. The algorithm was evaluated mainly on two tasks: precise time spike sequence generation, and spike pattern classification [12]. In spike sequence generation task, a spiking neuron is trained to generate any random spike train in response to a recognised pattern of input spike sequences (spike trains). This property is also used for temporal spike pattern classification by training the neuron to associate different spike trains to different input classes. Recently, we have extended the application of the algorithm to train multiple neurons to classify multiple classes of spike patterns generated artificially [10].

In this paper, we investigate SPAN learning on a practical dataset where the task is to classify images of handwritten digits from the MNIST dataset [7]. The first stage in the learning process is to convert the images into spike patterns. The conversion is done using the Virtual Retina [19] -a simulator that transforms image/video into pattern of spike trains. The generated patterns are used to train a single layer of SPANs for classification. The MNIST is a non-linear dataset which guarantees that the different spike pattern classes have more complicated inter/intra- class characteristics, i.e., patterns that belong to the same class are varied and there is overlap between different classes, giving a clear indication of the feasibility of SPAN learning for a practical applications.

2 Image to Spike Coding

The main function of the biological retina is converting input image stimulus to patterns of spikes. How different features of the stimulus relate to the structure of the generated spike pattern is not clear. A number of research works have proposed simplified software and hardware tools to model the retina and other parts of the visual system. Rank Order Coding (ROC) is one of the earliest coding scheme for image coding is based on the biological principle of fast processing of image stimulus in the brain [16]. Information is encoded in the order of spike firing across a population of neurons in which each neuron fires at most a single spike. Based on this coding, a face identification system is proposed [2] and also an evolving SNN (eSNN) architecture for integrated audio-visual information processing and pattern recognition [20].

A piece of hardware referred to Silicon Retina(SR) was made to convert input streams of image frames into spike patterns in a format referred to Address Encoding Representation (AER) [8]. The pixels of the SR respond to events that represent relative changes in intensity by computing the difference in pixel intensity between two successive frames (temporal contrast) and generating spikes if this difference exceeds a threshold value. Although the concept is simple, the hardware implementation provides fast computation power necessary for certain vision application.

The Virtual Retina (VR) is a software simulator that models more complicated aspects of the biological retina. It transforms a video input into spike patterns [19]. The VR consists of three stages of processing layers that correspond to different layers of the retina, namely the Outer Plexiform Layer (OPL), Contrast Gain Control (CGC) and the Ganglion layer (GL). Particularly, the GL layer is responsible of converting continuous current into spike trains. The conversion is performed using a Noisy Leaky Integrate and Fire (LIF) neuron, where the time delay of the generated spikes is proportional to the input current. The noise is represented as a random value added to the membrane potential of the LIF neuron in order to reproduce the variability found in trail-to-trail spike recording of real ganglion cells. Because the conversion is temporal, the VR is suitable to use for SPAN learning which is also based on temporal coding. We use a basic configuration of the VR, that consists of OPL and a single GL, as an encoder to convert digit images into spike patterns.

3 SPAN Learning Method and Network Topology

In this section, we describe briefly SPAN learning rule and the SNN network architecture. More details can be found in previous publications [10–12]. SPAN rule is a supervised learning method to associate input spike pattern to a target spike train by adjusting the weights of the input synapses according to the following formula:

$$\Delta w_i = \lambda \left(\frac{e}{2}\right)^2 \left[\sum_g \sum_f (|t_i^f - t_d^g| + \tau_s) e^{-\frac{|t_i^f - t_d^g|}{\tau_s}} - \sum_h \sum_f (|t_i^f - t_a^h| + \tau_s) e^{-\frac{|t_i^f - t_a^h|}{\tau_s}} \right] \tag{1}$$

where τ_s is the kernel function time constant, e is the exponential constant, t_i, t_a and t_d are the times of the input, actual output and target spikes respectively,.

According to this rule, the synaptic weight w is adjusted based on the precise time of the input, output and target spikes. Fig. 1 shows the configuration of the SNN network, for two classes, trained by the above rule.

The network could be used to classify multi-class spike patterns or to generate different spike trains. In [10], we have evaluated the network in classifying multiple categories of spike patterns generated artificially. The dataset was created by

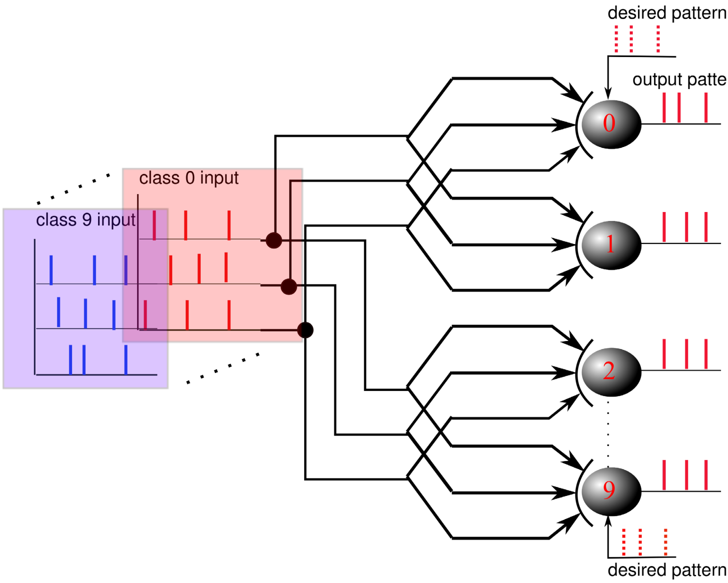


Fig. 1. The SNN topology used in this work. It is a single layer of spiking neurons, each neuron is trained to recognize a single digit.

generating a number of template spike patterns, based on a uniform distribution. Then, using these templates many samples were generated by adding time delay jitters to the spikes of the templates. Therefore, it is likely this procedure will lead to a dataset that is linearly separable. Hence, in next section we evaluate SPAN on a more realistic dataset.

4 Learning Handwritten Digits Using SPAN

4.1 Description of the data

The MNIST handwritten digits, available from [7], is a well-known dataset used by many researchers to evaluate pattern recognition methods. Each image is 28 × 28 pixels. We use 200 sample images per digit (a total of 2000 images) for training and the network is tested on different 200 images per digit. Each sample digit is converted into spike pattern using the Virtual Retina [19]. The produced spike pattern consists of 784 spike trains. Fig.2 shows an example of the generated spike patterns for four digits. It can be noted that it is quite difficult to recognize the digit by only looking at the spike pattern.

4.2 Experimental Setup

SPAN rule is used to train the network of Fig. 1 to learn the animated digit images. The network consists of ten neurons, each for one digit class. The weights

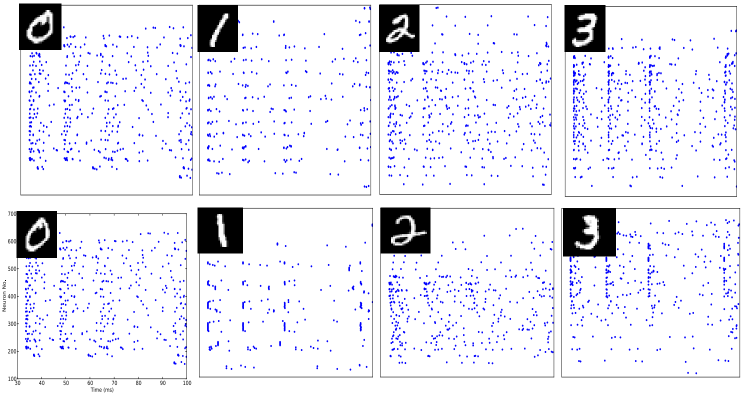


Fig. 2. Raster plots of the generated spike patterns by the Virtual Retina for the first four digits

are initialised randomly in the range [0.0, 5.0]. Each neuron has 784 synapses corresponding to (28 × 28) pixels of the input image. Therefore, there are 7840 synapses to be trained (784 synapse × 10 classes).

The neurons are Leaky Integrate-and-Fire (LIF) described by the following differential equation:

$$\tau_m \frac{du_i}{dt} = -u_i(t) + R\ I_i^{\mathrm{syn}}(t) \tag{2}$$

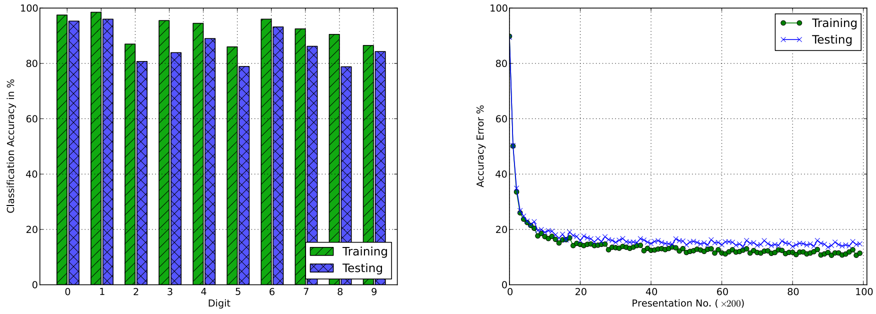
where I_i^{syn} is the input signal current. The constant $\tau_m = RC$ is called the membrane time constant of the neuron and fixed to 10ms. Whenever the membrane potential u_i crosses a threshold $\vartheta = 20$mv, the neuron fires a spike and its potential is reset to a reset potential $u_{\mathrm{reset}} = 0$mv. The learning rate (λ) in Eq. 1 is fixed to 0.01.

Each neuron is trained to produce a target spike train={25.,35.,45.,55.,65., 75., 85.,95.}ms selected randomly when a spike pattern from the assigned class is presented at the input and not to spike when patterns from other classes presented. In principle, different output spike patterns can be used for different digits.

The training is performed in 200 epochs, in each epoch the samples of each class (digit) are presented in random order. After each presentation of a training pattern, the synaptic weights of the neurons are updated according to Eq. 1. Thus, the training is performed in incremental mode [10] rather than batch mode, in which synapses are updated only after presenting the all training samples.

4.3 Results

We report the ability of the network to learn and classify the digit dataset in terms of classification accuracy. The classification accuracy on each digit class is defined as the number of patterns classified correctly over the total number of training(testing) patterns for that digit. Fig. 3a reports the obtained accuracy for the ten digits.



(a)

(b)

Fig. 3. (**a**) The average accuracy obtained in the training and testing phase for the ten digits. (**b**) Evolution of the classification error, computed using 200 training and testing patterns, after each training sample presentation.

The network was able to learn to recognize the ten digits with an average accuracy of 92% in the training set and 86.6% in the testing set. Digit 8 has the minimum accuracy of 78.8% while the highest accuracy was obtained for digit 1 with a value of 96%, i.e., digit 1 has the most distinctive spike patterns. The obtained results confirm the ability of the network to classify the digit dataset with a good efficiency. We note that we have obtained the same generalization(testing) accuracy of 86% when the trained network was evaluated on more testing samples (10000 testing samples).

To understand the network better, the evolving of the network performance during training is investigated. During the training phase, the misclassification accuracy(error), computed on 200 images from the training set and another 200 images from the testing set, are recorded after each input presentation. There are 2000 training samples and 200 epochs which leads to 2000 × 200 presentations. However, we report the testing and training misclassification error in a step of 200 and up to 40000 presentations as shown in Fig. 3b. The figure shows the testing error curve is following closely the training curve. After about 4000 presentation , which is equivalent to two training epochs, the error curves start to flat and show very slow change. Thus, it is possible to train the network with less than 20 epochs to obtain good results. In fact, there should be a balance between the number of training epochs and number of training samples. Sufficient number of training samples with few tens of epochs training are required for satisfactory training for this experiment.

5 Conclusion and Future Work

In this paper we have demonstrated the application of SNN trained with SPAN [10–12] on learning and classifying images of handwritten digits. One crucial factor in using SNN for real-world computer application is properly encoding the information into spike patterns. SPAN learning method is based on temporal coding, i.e., information is coded into the precise time of the spikes. Using the VR [19], it was possible to spike encode the digit images and classify the generated spike patterns efficiently. It is noted that the neurons are trained to produce the desired spike sequence in response to their class and not to fire for other classes. After training, the synaptic weights take positive (excitatory) and negative (inhibitory) values This means the neuron is learning its class and also learning to reject other classes through weights adjustment. It might be more desirable to design a mechanism that is based on inhibition to suppress the neuron firing, for example using a similar mechanism to "winner-takes-all" as it is in [1]. In addition, the used network consists of a single layer of spiking neurons and there are no specific features extracted for classification. Therefore, there is a space for more investigation to enhance the architecture of the network to achieve better results. Although the digit images are static data where other conventional methods can perform better, however after spike encoding the generated spike patterns are temporal data. The video data will take a similar form after spike conversion, indicating that the proposed method has also the potential to be applied for video signals with little modification to the algorithm, which will be our future work to investigate. Furthermore, SPAN learning will be enhanced for online learning and classification. We are also investigating the feasibility of SPAN implementation on a SNN chip scuh as the SRAM- based chip [13]. Such implementation will make it possible to use SPAN for a broad range of engineering applications based on the principle of embedded systems.

Acknowledgement. This work is supported by the Knowledge Engineering and Discovery Research Institute (www.kedri.info) of Auckland University of Technology. GL is also supported by an EU exchange grant from the University of Trento. NK is also supported by EU Marie-Curie IIF funded project 'EvoSpike' (http://ncs.ethz.ch/projects/evospike/.

References

1. Brader, J.M., Senn, W., Fusi, S.: Learning real-world stimuli in a neural network with spike-driven synaptic dynamics. Neural Comput. 19(11), 2881–2912 (2007)
2. Delorme, A., Gautrais, J., van Rullen, R., Thorpe, S.: Spikenet: A simulator for modeling large networks of integrate and fire neurons. Neurocomputing 26-27, 989–996 (1999)
3. Delorme, A., Thorpe, S.J.: Face identification using one spike per neuron: resistance to image degradations. Neural Networks 14(6-7), 795–803 (2001)
4. Floreano, D., Epars, Y., Zufferey, J.C., Mattiussi, C.: Evolution of spiking neural circuits in autonomous mobile robots: Research articles. Int. J. Intell. Syst. 21(9), 1005–1024 (2006)
5. Gerstner, W., Kistler, W.M.: Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge (2002)
6. Hopfield, J.: Pattern recognition computation using action potential timing for stimulus representation. Nature 376, 33–36 (1995)

7. LeCun, Y., Cortes, C.: The mnist database of handwritten digits (1998), http://yann.lecun.com/exdb/mnist/
8. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. In: IEEE International Solid-State Circuits Conference, ISSCC 2006. Digest of Technical Papers, pp. 2060–2069 (2006)
9. Maass, W.: Networks of spiking neurons: The third generation of neural network models. Neural Networks 10(9), 1659–1671 (1997)
10. Mohemmed, A., Kasabov, N.: Incremental learning algorithm for spatio-temporal spike pattern classification. In: IEEE World Congress on Computational Intelligence, WCCI 2012, Brisbane, Australia, pp. 1227–1232 (2012)
11. Mohemmed, A., Schliebs, S., Matsuda, S., Kasabov, N.: Method for Training a Spiking Neuron to Associate Input-Output Spike Trains. In: Iliadis, L., Jayne, C. (eds.) EANN/AIAI 2011, Part I. IFIP AICT, vol. 363, pp. 219–228. Springer, Heidelberg (2011)
12. Mohemmed, A., Schliebs, S., Matsuda, S., Kasabov, N.: Span: Spike pattern association neuron for learning spatio-temporal spike patterns. International Journal of Neural Systems 22(04), 1250012 (2012)
13. Moradi, S., Indiveri, G.: A vlsi network of spiking neurons with an asynchronous static random access memory. In: 2011 IEEE Biomedical Circuits and Systems Conference (BioCAS), pp. 277–280 (2011)
14. Nikolic, D., Hausler, S., Singer, W., Maass, W.: Distributed fading memory for stimulus properties in the primary visual cortex. PLoS Biol. 7(12), e1000260 (2009)
15. Panchev, C., Wermter, S.: Temporal sequence detection with spiking neurons: Towards recognizing robot language. Instruction, Connection Science 18, 1–22 (2006)
16. Thorpe, S.J.: Spike arrival times: A highly efficient coding scheme for neural networks. In: Eckmiller, R., Hartmann, G., Hauske, G. (eds.) International Conference on Parallel Processing in Neural Systems, pp. 91–94. Elsevier, North-Holland (1990)
17. Uysal, I., Sathyendra, H., Harris, J.: Towards spike-based speech processing: A biologically plausible approach to simple acoustic classification. Int. J. Appl. Math. Comput. Sci. 18, 129–137 (2008)
18. Van Rullen, R., Gautrais, J., Delorme, A., Thorpe, S.: Face processing using one spike per neurone. Biosystems 48(1-3), 229–239 (1998)
19. Wohrer, A., Kornprobst, P.: Virtual retina: A biological retina model and simulator, with contrast gain control. Journal of Computational Neuroscience 26, 219–249 (2009)
20. Wysoski, S.G., Benuskova, L., Kasabov, N.: Evolving spiking neural networks for audiovisual information processing. Neural Networks 23(7), 819–835 (2010)

DPSO Based on Random Particle Priority Value and Decomposition Procedure as a Searching Strategy for the Evacuation Vehicle Routing Problem

Marina Yusoff[1], Junaidah Ariffin[2], and Azlinah Mohamed[1]

[1] Intelligent System Research Group
Faculty of Computer and Mathematical Sciences
Universiti Teknologi Mara
40450 Shah Alam, Selangor
Malaysia
{marinay,azlinah}@tmsk.uitm.edu.my
[2] Flood-Marine Excellence Centre
Faculty of Civil Engineering
Universiti Teknologi Mara
40450 Shah Alam, Selangor
Malaysia
junaidahariffin@yahoo.com

Abstract. Flood evacuation operations face a difficult task in moving affected people to safer locations. Uneven distributions of transport, untimely assistance and poor coordination at the operation level are among the main problems in the evacuation process. This is attributed to the lack of research focus on evacuation vehicle routing. This paper proposes an improved discrete particle swarm optimization (DPSO) with a random particle priority value and decomposition procedure as a searching strategy to solve evacuation vehicle routing problem (EVRP). The search strategies are proposed to reduce the searching space of the particles to avoid local optimal problem. This algorithm was computationally experimented with different number of potentially flooded areas, various types of vehicles, and different speed of vehicles with DPSO and genetic algorithm (GA). The findings show that an improved DPSO with a random particle priority value and decomposition procedure is highly competitive. It offers outstanding performance in its fitness value (total travelling time) and processing time.

Keywords: Decomposition Procedure, Discrete Particle Swarm Optimization, Evacuation Vehicle Routing Problem; Priority Value, Potentially Flooded Area.

1 Introduction

Floods can be defined as the event resulting from heavy rainfall over a small area within a short time that can cause water to rise and fall quite rapidly [1]. The occurrence of floods is unpredictable. These instantaneous events usually occur due to inconsistent weather patterns, series of storm and extraordinary rainfall [2]. A lot of

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 678–685, 2012.

people have to be evacuated at the shortest possible time to avoid loss of lives. In Malaysia, uneven distributions of transport, untimely assistance and poor coordination at the operation level have always been the main problem in the evacuation process during a flood event. Planning for evacuation is vital in assisting people to move to a safe place. As time is a decision factor in the evacuation process, urgent and firmly decisions are very much required at the operational level [3]. An evacuation plan should be efficiently constructed by taking into account routes for vehicles. Thus, routing the vehicles to potentially flooded area (PFA) is one of the primary concerns in the evacuation process.

Several studies on evacuation plan applied optimization approaches on different types of disasters namely capacity constrained route planning [4], A* [5], Flip High Flip Edge (FHFE) [6], greedy heuristic [7][8], SP-TAG [9], ant colony optimization (ACO) [10], route construction heuristic and local search [11], particle swarm optimization (PSO) [12], and hybrid genetic algorithm (GA), and simulated annealing (SA) [13] have demonstrated good performance. However, research on the evacuation route problem for an optimal solution is pertinent. Vehicle travelling time has to be reduced to ensure all people arrive safely at destination. Thus, this paper addresses this gap of which route of vehicles is addressed and enhanced the work of [14] and [15] on evacuation vehicle routing problem (EVRP).

The remainder of this article is organized as follows. Section 2 describes the EVRP and its features. Section 3 presents two solution representation and two DPSO algorithms for the EVRP solutions. The parameters, datasets and computational results are discussed in Section 4, and finally, Section 5 concludes the paper and opens some lines for future research.

2 The Evacuation Vehicle Routing Problem

This section explains the steps involved in finding a solution for EVRP comprising of EVRP formulation, the solution representation, and DPSO algorithms.

2.1 Problem Formulation

The EVRP involves a static routing of a number of vehicles from vehicle location multiple PFA. EVRP addresses the objective function to find the minimum total travelling time for all vehicles from vehicle location to the PFA. The problem can be formally defined as follows:

Let $G = (N, E)$ be a weighted directed graph. Define $N = \{N_0, N_1, \ldots, N_n\}$. N_0 represents the vehicle location and N_n is the destination node (PFA). E is the set of edges. t_{ij} represents the travelling cost of traversing from i to j. For each edge $(i, j) \in E$, travel time $t_{ij} \geq 0$, is a non negative integers. $H = \{H_1, H_2, \ldots, H_k\}$ is the set of all vehicles that are able to move from node i and j. The objective function is to find the minimum total travelling time for all vehicles from N_0 to N_n. The EVRP is mathematically formulated as shown below:

$$\text{Minimize} \sum_i^n \sum_j^n \sum_k^m T_{ijk} X_{ij} \tag{1}$$

Subject to:

$$\sum_j^n X_{ij} - \sum_{k=1}^n X_{ki} = \begin{cases} 1 & if\ i = 0 \\ 0 & if\ i = 1, \dots, n-1 \\ -1 & if\ i = n \end{cases} \tag{2}$$

$$X_{ij} \in \{0,1\}, (i,j) \in E \tag{3}$$

where:

I = index of nodes, $i \in$N
j = index of nodes, $j \in$N
k = index of vehicle H, $k \in H$
T_{ijk} = travelling time of vehicle k traversing from i to j.
X_{ij} =binary variable which is 1 if node i to node j *is* traversed, otherwise it is 0.

Constraints 2 ensure that the path starts at N_0, end at N_n, and either pass through or avoid every other node j. Constraint 3 is the set of bound decision variables. The solution representation for EVRP was adopted from the work of Mohammed et al. [14] because of its good performance in solving SPP. To accommodate the EVRP, this solution representation is enhanced and takes into account a number of the vehicles. However, the use of PV that represents each node is maintained. The solution representation described above was enhanced from [14] and embedded in the myDPSO_1 algorithm. The new solution for EVRP has adopted the similar process of nodes expansion and the random selection of PV as stated in [15].

2.2 myDPSO_2 Algorithm

The new solution representation for EVRP that is discussed above is implemented in myDPSO_2 algorithm. The algorithm starts with the normal process of PSO. Step 2 and 3 initialize the number of population and the coefficient values C_1 and C_2, respectively. Step 4 performs the initialization of PV and velocities. Step 5 retrieves vehicle's information which includes the vehicle id, vehicle capacity, and its standard travelling speed. Step 6 and 7 perform the search decomposition procedure for each of the vehicle. In this step, only one path is selected and the selected node is assigned with PV_{min} upon selection of the path as demonstrated in Fig. 2. The *Pbest* and *Gbest* of each particle are calculated upon expansion of all nodes. *Pbest* is the total distance for each particle, whereas *Gbest* is the minimum total distance obtained from all particles. The iteration process starts at step 9 through 22 until a maximum iteration is achieved. In this iteration, each particle is updated with a new velocity and new position value (PV) at step 10 until 12. The new velocity and position value are in the form of the positive integer. Then, PV for all sub particles is updated using step 13. Step 14 performs the decomposition procedure of PV. $Pbest_{(new)}$ and $Gbest_{(new)}$ are calculated at step 15 and 16, respectively. Finally, steps 17 through 21 are the conditions for the selection of the best current fitness for each of the iteration.

myDPSO_2 algorithm

```
1:      Begin
2:          Initialize number of   population
3:          Declare   $C_1$ and $C_2$
4:          Initialize PV, $V_{intialize(min)}$  and $V_{initialize(max)}$for all particles in random
5:          Retrieve vehicle's information from [15]
6:          For each vehicles
7:              Perform search decomposition procedure
8:              Calculate Pbest and Gbest
9:          Do
10:         For each particle
11:             Calculate $V_{(new)}$
12:             Calculate $PV_{(new)}$
13:             Update PV for all sub particles
14:             Perform step 6 and 7
15:             Calculate Pbest $_{(new)}$
16:             Calculate Gbest $_{(new)}$
17:             If (Gbest $_{(new)} >$ Gbest)
18:                 Assign $G_{best}$ as the best current fitness
19:             If (Gbest $_{(new)} =<$ Gbest)
20:                 Gbest= Gbest $_{(new)}$
21:             Assign Gbest$_{(new)}$ as the best current fitness
22:     While (maximum iteration is achieved)
23: End
```

3 Performance of Algorithms Using Multiple PFA

Parameters are applied and the value of inertia weight is selected from the range of this parameter suggested by Shi and Eberhart [17]. The stopping condition is based on all vehicles are arrived destination or 200 iterations and 30 experiments were done for each of the datasets. Datasets were taken from a flash flood in Malaysia's Kota Tinggi district in 2007. Routes from vehicle location to PFA are indicated by source nodes (original vehicle location), nodes, edges, and destination node (PFA). All routes are transformed into graph abstraction. The graph is then transferred into an adjacency matrix for easy transformation into the algorithm. Table 1 shows the datasets for routing comprising the number of nodes, the total number of people that need to be evacuated and the number of vehicles generated based on [18].

Table 1. List of datasets from flash flood evacuation in 2006 and 2007 for a single PFA

Datasets	Number of nodes	Number people	Number of vehicles
VR1_PFAs_07	49	1566	238
VR2_PFAs_07	61	3106	374
VR3_PFAs_07	88	3180	355
VR4_PFAs_07	109	3800	496
VR5_PFAs_07	133	3996	516

The following tables show the computational results for multiple PFA. Results for myDPSO_2 performed better than the other three algorithms as shown in Table 2. The total travelling time for GA_2 is slightly lower than myDPSO_2. Contrary to expectations, neither myDPSO_1 nor GA_1 produced any result after 200 iterations for VR1_PFAs_07. This failure may be attributed to the fact that the multi-valued PV assigned to each node failed to determine the valid paths. The dataset used greater number of nodes than a single PFA. This shows that the particles in multiple PFA utilize better the search space compared to a single PFA.

Table 2. Performance of DPSOs and GAs using VR1_PFAs_07

	myDPSO_1		GA_1		myDPSO_2		GA_2	
	tt_{vs}	*PT (s)*	*tt_{vs}*	*PT (s)*	*tt_{vs}*	*PT (s)*	*tt_{vs}*	*PT (s)*
Avg	-	-	-	-	10.170	**3.314**	10.173	**3.452**
Min	-	-	-	-	10.167	0.920	10.167	1.482
Max	-	-	-	-	10.187	5.132	10.193	8.596
Std Dev	-	-	-	-	0.009	1.085	0.007	1.565

* tt_{vs} – total travelling time (hour), PT - processing time (second), iter - number of iteration.

The next comparison highlights (Table 3) the results of VR2_PFAs_07. It was found that myDPSO_2 produced the best solution quality and used less processing time. However, myDPSO_1 and GA_1 failed to obtain any results. myDPSO_1 stopped at 26th iteration, whereas GA_1 at 27th iteration. They failed to iterate up to 200 iterations. With an increase in the number of vehicles and number of PFA for this dataset involving 374 vehicles to travel from the vehicle location to three PFA, more computer memory is required to converge. This may be because of many arrays in Java coding and the use of different structure of coding in the language which is not in the scope of this study. So far, the proposed myDPSO_2 has shown good results with only one iteration for convergence (all vehicles arrive at the assigned PFA). As can be seen in 1, the average processing time of myDPSO_2 is slightly higher than GA_2 for VR2_PFAs_07. The total travelling time for myDPSO_2 is 0.29% lower than GA_2, which is about 1.98 minutes. Hence, the result has confirmed the objective function that was stated in the problem formulation of EVRP. With a minimum total travelling time, all people can be picked-up by the assigned vehicles at each of the PFA at the shortest time.

Table 3. Performance of DPSOs and GAs using VR2_PFAs_07

	myDPSO_1		GA_1		myDPSO_2		GA_2	
	tt_{vs}	*PT (s)*	*tt_{vs}*	*PT (s)*	*tt_{vs}*	*PT (s)*	*tt_{vs}*	*PT (s)*
Avg	-	-	-	-	**11.494**	**9.358**	11.527	**9.095**
Min	-	-	-	-	11.385	3.978	11.385	4.493
Max	-	-	-	-	11.723	22.433	13.583	21.419
Std Dev	-	-	-	-	0.090	3.826	0.391	4.141

* tt_{vs} – total travelling time (hour), PT - processing time (second), iter - number of iteration.

As shown in Table 4, myDPSO_1 and GA_1 failed to obtain any results, myDPSO_1 stopped at only 19th iteration. Meanwhile, GA_1 run at 10th iteration. The result validates the employment of myDPSO_2 in solving EVRP. This algorithm provides results with less total processing time compared to GA_2 to move all vehicles from vehicle location to four PFAs, using dataset of VR3_PFAs_07. This result again ensures all vehicles arrive at PFA at a minimum total travelling time, which is important in evacuation planning.

Table 4. Performance of DPSOs and GAs using VR3_PFAs_07

	myDPSO_1		GA_1		myDPSO_2		GA_2	
	tt_{vs}	*PT (s)*	tt_{vs}	*PT (s)*	tt_{vs}	*PT (s)*	tt_{vs}	*PT (s)*
Avg	-	-	-	-	**12.170**	**7.594**	12.174	**17.006**
Min	-	-	-	-	11.736	4.025	11.625	3.931
Max	-	-	-	-	13.466	22.479	13.257	309.620
Std Dev	-	-	-	-	0.535	3.836	0.552	55.332

* tt_{vs} – total travelling time (hour), PT - processing time (second), iter - number of iteration.

It is similar to VR3_PFAs_07, VR4_PFAs_07 and VR5_PFAs_07 are also failed to obtain any results. On the other hand, myDPSO_2 outperformed GA_2 for both of VR4_PFAs_07 and VR5_PFAs_07 in its total travelling time and processing time. It is noted that the use of myDPSO_2 has successfully achieved the best performance among the other three algorithms. Thus, these results confirmed that this algorithm satisfy the objective function which is to find the minimum total travelling time.

Table 5 shows that there was a significant difference for fitness value (*p*-value = 0.008) for a pair of myDPSO_2 and GA_2. However, no significant difference is noted for processing time. The finding validates the consistent performance obtained by myDPSO_2.

Table 5. A pair t-test results for myDPSO_2 and GA_2 for the multiple PFA datasets

	Pair	T	df	Sig. (2-tailed)
Fitness value	myDPSO_2 - GA_2	-2.672	299	.008
Processing time	myDPSO_2- GA_2	-.904	299	.367

4 Conclusions

This paper presents the solution to EVRP in achieving the objective function which is to find the minimum total travelling time for all the vehicles from vehicle location to the PFA. The solution representation and a modified solution representation for the EVRP solution were addressed. The solution representation is embedded in myDPSO_1 while the modified solution representation is embedded in myDPSO_2. They were compared to GA_1 that embedded solution representation and GA_2 which embedded a modified solution representation. myDPSO_2 was found to be the suitable algorithm for solving EVRP for multiple PFA. Thus, it can be said that the search decomposition procedure with random selection of PV embedded in myDPSO_2 and GA_2 provided better solutions compared to the solution that was embedded in myDPSO_1 and GA_1. Overall, it can be concluded that myDPSO_2 that was applied with a new solution representation provided better results compared to GA_2, myDPSO_1, and GA_1 for multiple PFA datasets based on the t-test evaluation. Further experiments are required considering different parameter settings and large size of EVRP datasets.

Acknowledgment. Authors acknowledge fund from the Ministry of Science and Technology Malaysia through the Science Fund and University Technology MARA.

References

1. Barredo, J.I.: Major flood disasters in Europe: 1950–2005. Natural Hazards 42(1), 125–148 (2007)
2. Shafie, A.: A Case Study on Floods of 2006 and 2007 in Johor, Malaysia. Colorado State University (2009)
3. Yusoff, M., Ariffin, J., Mohamed, A.: A Modified Discrete Particle Swarm Optimization for Solving Flash Floods Evacuation Operation. International Journal of Computers 5(4), 460–467 (2011)
4. Lu, Q., George, B., Shekhar, S.: Capacity Constrained Routing Algorithms for Evacuation Planning: A Summary of Results. In: Proceedings of 9th International Symposium on Spatial and Temporal Databases, pp. 291–307 (2005)
5. Lu, Q.: Capacity constrained routing algorithms for evacuation route planning. PHD Thesis, University of Minnesota (2006)
6. Kim, S., Shekhar, S.: Contraflow network reconfiguration for evacuation planning: a summary of results. In: 13th Annual ACM International Workshop on Geographic Information Systems, pp. 250–259 (2005)
7. Shekhar, S., Kim, S.: Contraflow transportation network reconfiguration for evacuation route planning. Technical Report, Mn/DOT 2006-21, Department of Computer Science and Engineering, University of Minnesota (2006)
8. Kim, S., Shekhar, S., Min, M.: Contraflow transportation network reconfiguration for evacuation route planning. IEEE Transactions on Knowledge and Data Engineering 20(8), 1115–1129 (2008)
9. George, B., Kim, S., Shekhar, S.: Spatio-temporal Network Databases and Routing Algorithms: A Summary of Results. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 460–477. Springer, Heidelberg (2007)
10. Yi, W., Kumar, A.: Ant Colony Optimization for Disaster Relief Operations. Transportation Research Part E: Logistics and Transportation Review 43(6), 660–672 (2007)
11. Abdelgawad, H., Abdulhai, B.: Managing Large-Scale Multimodal Emergency Evacuations. Journal of Transportation Safety & Security 2(2), 122–151 (2009)
12. Wang, J.W., Ip, W.H., Zhang, W.J.: An integrated road construction and resource planning approach to the evacuation of victims from single source to multiple destinations. IEEE Transactions on Intelligent Transportation System 11(2), 277–289 (2010)

13. Xie, C., Turnquist, M.A.: Lane-based evacuation network optimization: An integrated Lagrangian relaxation and tabu search approach. Transportation Research Part C: Emerging Technologies 19(1), 40–63 (2011)
14. Mohemmed, A.W., Sahoo, N.C., Geok, T.K.: Solving shortest path problem using particle swarm optimization. Applied Soft Computing 8(4), 1643–1653 (2008)
15. Yusoff, M., Ariffin, J., Mohamed, A.: A Modified Discrete Particle Swarm Optimization for Solving Flash Floods Evacuation Operation. Journal of Computers 5(4), 460–467 (2011)
16. Shi, Y., Eberhart, R.: A Modified Particle Swarm Optimizer. In: Proceeding of the 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence, pp. 69–73 (1998)
17. Yusoff, M., Ariffin, J., Mohamed, A.: Solving Vehicle Assignment Problem Using Evolutionary Computation. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010, Part I. LNCS, vol. 6145, pp. 523–532. Springer, Heidelberg (2010)

A Quantum-Inspired Evolutionary Algorithm for Optimization Numerical Problems

Maurizio Fiasché

MOX – Department of Mathematics "F. Brioschi",
Polytechnic Institute of Milan, Milan, Italy
maurizio.fiasche@ieee.org

Abstract. This paper proposes a novel type of quantum-inspired evolutionary algorithm (QiEA) for numerical optimization inspired by the multiple universes principle of quantum computing, which is based on the concept and principles of quantum computing, such as a quantum bit and superposition of states. Numerical optimization problems are an important field of research with several applications in several areas: industrial plant optimization, data mining and many others, and although being successfully used for solving several optimization problems, evolutionary algorithms still present issues that can reduce their performances when faced with task where the evaluation function is computationally intensive. In order to address those issues the QiEA represent the most recent advance in the field of evolutionary computation. This work present some application about combinatorial and numerical optimization problems.

Keywords: Quantum Computing, Quantum Inspired, Evolutionary Algorithms, Optimization Problems.

1 Introduction

Evolutionary algorithms (EAs) are principally a stochastic search and optimization method based on the principles of natural biological evolution. EAs operate on a population of potential solutions, applying the principle of 'survival of the fittest' to produce successively better approximations to a solution. At each generation of the EA, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and reproducing them using variation operators. This process may lead to the evolution of populations of individuals that are better suited to their environment than the individuals from which they were created, just as in natural adaptation. EAs are characterized by the representation of the individual, the evaluation function representing the fitness level of the individuals, and the population dynamics such as population size, variation operators, parent selection, reproduction and inheritance, survival competition method, etc. To have a good balance between exploration and exploitation, these components should be designed properly. In particular, in this paper the representation and population dynamics are investigated to represent the individuals effectively to explore the search space with a smaller number of individuals (even with only one individual for real-time application) and to exploit the search space for

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 686–693, 2012.

a global solution within a short span of time, respectively. For these purposes, some concepts of quantum computing are adopted in this paper for presenting the proposed evolutionary algorithm. Quantum computing is a research area which includes concepts like quantum mechanical computers and quantum algorithms. Quantum mechanical computers were proposed in the early 1980s [1] and their description was formalized in the late 1980s [2]. Many efforts on quantum computers have progressed actively since the early 1990s because these computers were shown to be more powerful than digital computers for solving various specialized problems. There are well-known quantum algorithms such as Deutsch-Jozsa algorithm [3], Simon's algorithm [4], Shor's quantum factoring algorithm [5], and Grover's database search algorithm [6]. For giving an idea: if, for example, the speed of quantum or digital computer is 1 MIPS, Grover's algorithm can find the secret key of 56-bit string within 4 minutes in quantum computer without any factoring algorithms, while the classical algorithm can find it within 1000 years [7]. In particular, since the difficulty of the factoring problem is crucial for the security of the RSA cryptosystem which is in widespread use today, interest in quantum computing is increasing. Research on merging evolutionary computation and quantum computing has started since the late 1990s. Here, the concept of quantum computing utilizes the special nonlocal properties of the quantum phenomena. A quantum atomic or subatomic particle (e.g. atoms, electrons, protons, neutrons, bosons, fermions, photons) exists in a probabilistic superposition of states rather than in a single definite state. Particles in general are characterized by: charge, spin, position, velocity, and energy [8]. In this paper we want to propose a novel EA for numerical optimization inspired by the multiple universes principle of quantum computing that presents faster convergence time for the benchmark problems. In section one we introduce Evolutionary theory and its application for optimization problems, in section two we give an introduction for quantum principles, in section three structure of QiEA is described and in section four a customizing of the algorithm is proposed and results are discussed, finally in section 5 some conclusions and points toward future works are presented.

2 Evolutionary Strategy: An Overview

EAs are based on computational models of fundamental evolutionary processes such as selection, recombination and mutation. Individuals, or current approximations, are encoded as strings composed over some alphabet(s), e.g. binary, integer, real-valued, etc., and an initial population is produced by randomly sampling these strings. Once a population is produced, it may be evaluated using an objective function which characterizes an individual's performance in the problem domain. The objective function is also used as the basis for selection, and determines how well an individual performs in its environment. Genetic algorithms emphasize recombination (crossover) as the most important search operator and apply mutation with very small probability solely as a background operator. They also use a probabilistic selection operator (proportional selection) and often rely on a binary representation of individuals [9].

Evolution strategies use normally-distributed mutations to modify real-valued vectors and emphasize mutation and recombination as essential operators for searching in the search space and in the strategy parameter space at the same time. The selection operator is deterministic, and parent and offspring population sizes usually differ from each other [9]. Evolutionary programming emphasizes mutation and does not incorporate the recombination of individuals. Similarly to evolution strategies, when approaching real-valued optimization problems, evolutionary programming also works with normally distributed mutations and extends the evolutionary process to the strategy parameters. The selection operator is probabilistic. Presently, most applications are reported for search spaces involving real-valued vectors, although the algorithm was originally developed to evolve finite-state machines [10].

3 Quantum Theory Principles

Han and Kim proposed a Quantum Evolutionary Algorithm (QEA) in 2002 [11], which was inspired by the concept of quantum computing. According to the classical computing concept, the smallest information unit in today's digital computers is one *bit*, existing as state '1' or '0' at any given time. The corresponding analogue in a quantum inspired representation is the quantum bit (*qbit)* [12]. Similar to classical bits a *qbit* may be in '1'or '0' states, but also in a *superposition* of both states. Superposition allows the possible states to represent both 0 and 1 simultaneously based on its probability. A qbit state $|\Psi>$ can be described as:

$$|\Psi> = \alpha|0> + \beta|1> \qquad (1)$$

where α and β are complex numbers that are used to define the probability of which of the corresponding states is likely to appear when a *qbit* is read (measured, collapsed). $|\alpha|^2$ and $|\beta|^2$ give the probability of a qbit being found in state '0' or '1' respectively. Normalization of the states to unity guarantees:

$$|\alpha|^2 + |\beta|^2 = 1 \qquad (2)$$

at any time. The *qbit* is not a single value entity, but is a function of parameters which values are complex numbers. In order to modify the probability amplitudes, *quantum gate operators* can be applied to the states of a *qbit or a qbit* vector. A quantum gate is represented by a square matrix, operating on the amplitudes α and β in a Hilbert space, with the only condition that the operation is reversible. Such gates are: NOT-gate, rotation gate, Hadamard gate, and others [12]. General notation for an individual with several qubits can be defined as:

$$Q(t) = \{q^t_1, q^t_2,\ldots,q^t_n\} \qquad (3)$$

where n is the size of the population and q^t_j is a Q-bit individual.

Another quantum principle is *entanglement* - two or more particles, regardless of their location, can be viewed as "correlated", undistinguishable, "synchronized", coherent. If one particle is "measured" and "collapsed", it causes for all other entangled particles to "collapse" too. Two fundamental motivations for the development of EAs that utilize quantum computation are:

i. The properties of a quantum representation of probabilities;
ii. The recent manifestation that quantum inspired evolutionary algorithms (QiEA) are probability estimation of distribution algorithms (EDA) [8].

4 Quantum Inspired Evolutionary Algorithms

Inspired by the concept of quantum computing, QiEA is designed with a Q-bit representation, a Q-gate as a variation operator, and an observation process. The representation, the proposed algorithm, and its characteristics are described in the following.

A number of different representations can be used to encode the solutions onto individuals in evolutionary computation. The representations can be classified broadly as binary, numeric, and symbolic [12]. QiEA uses the representation reported above, called a Qbit, for the probabilistic representation that is based on the concept of qubits, and a Q-bit individual as a string of Q-bits, which are defined above and in [11]. Evolutionary algorithm with Q-bit representation has a better characteristic of population diversity than any other representations, since it can represent linear superposition of states probabilistically.

4.1 Structure of QiEA

QiEA is a probabilistic algorithm similar to other evolutionary algorithms. QiEA, however, maintains a population of Q-bit individuals, where Q(t) at generation *t*, where *n* is the size of population, and q^t_j is a Q-bit individual defined as

$$\left[\begin{array}{c|c|c|c}\alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_2 & \beta_2 & \cdots & \beta_m\end{array}\right]$$

where the following holds for i = 1, 2, …, m as described previously, i.e., the string length of the Q-bit individual, and $|\alpha|^2 + |\beta|^2 = 1$, $j = 1,2, \ldots, n$. Figure 1 and 2 show the procedure QiEA and the overall structure of QiEA explained in detail in [11][13], we want to specify some points useful for better comprising the rest of the paper:

1) In the step of 'initialize *Q*(*t*)' all q^t_j are initialized with 1/*V*2. It means that one Q-bit individual, q^0_j represents the linear superposition of all the possible states with the same probability:

$$|\psi_{q^0_j}> = \sum_{k=1}^{2^m} \frac{1}{\sqrt{2^m}} |X>$$

where X_k is the *k*th state represented by the binary string $(x_1, x_2 \ldots x_m)$, where x_i, $i = 1, 2, \ldots, m$, is either 0 or 1 according to the probability of either $|\alpha^0_i|^2$ or $|\beta^0_i|^2$, respectively. However, it should be noted that the performance of QEA can be influenced by the initial value.

2) Each binary solution x0*j* is evaluated to give a measure of its fitness.

3) The initial best solutions are then selected among the binary solutions *P*(0), and stored into *B*(0), where $B(0) = \{b^0_1, b^0_2, \ldots, b^0_n\}$ and b^0_j ($b^0_j|_{t=0}$) is the same as x^0_j at the initial generation.

4) Until the termination condition is satisfied, QiEA is running in the **while** loop. In particular, termination criteria are described in the next section .
5) In this step, Q-bit individuals in *Q*(*t*) are updated by applying Q-gates defined following rotation gate used as a basic Q-gate in QEA[13]
6, 7) The best solutions among *B*(*t* - 1) and *P*(*t*) are selected and stored into *B*(*t*), and if the best solution stored in *B*(*t*) is better fitted than the stored best solution b, the stored solution b is replaced by the new one.
8, 9) If the global migration condition is satisfied, the best solution b is migrated to *B*(*t*) globally. If the local migration condition is satisfied, the best one in a local group in *B*(*t*) is migrated to others in the same local group. The migration process defined below can induce a variation of the probabilities of a Q-bit individual.

Procedure QiEA

```
begin
t ← 0
1)  initialize Q(t)
      make P(t) by observing the states of Q(t)
2)  evaluate P(t)
3)  store the best solutions among P(t) into B(t)
4)  while (not termination condition) do
begin
t ← t + 1
vi)     make P(t) by observing the states of Q(t - 1)
vii)    evaluate P(t)
5)    update Q(t) using Q-gates
6)    store the best solutions among B(t - 1) and P(t)
into B(t)
7)     store the best solution b among B(t)
8)     if (global migration condition)
then migrate b to B(t) globally
9)    else if (local migration condition)
then migrate $b^{t}_{j}$ in B(t) to B(t) locally
end
end
```

Fig. 1. Procedure QiEA

5 Customization of QiEA

QiEA was first discussed in [11][13] and there are several variants of QiEA The main idea of QiEA is to use a standard EA function to update the particle position represented in a qubit. Here, we have applied a new variant to the algorithm that gives in our test a better performance of classical QEA [12]. We have applied an initial value of Han and applied a customization in termination criteria. Our proposal is that H_ε gate is used as a Q-gate, the classical termination conditions of

$$C_{av} = \left(\frac{1}{n} \sum_{j=1}^{n} C_b(q_j) \right) > \gamma$$

and

$$C_{\max} = \left(\max_{j=1}^{n} C_b(q_j) \right) > \gamma$$

should be modified as

$$C_{av} = \left(\frac{1}{n} \sum_{j=1}^{n} C_b(q_j) \right) > \left(1 - \frac{1}{2}\varepsilon + \frac{1}{3}\varepsilon^2 \right) \gamma$$

and

$$C_{\max} = \left(\max_{j=1}^{n} C_b(q_j) \right) > \left(1 - \frac{1}{2}\varepsilon + \frac{1}{3}\varepsilon^2 \right) \gamma$$

respectively. To increase the period for fine tuning caused by the ε boundary, the mixed termination condition can be also used as follows: MAXGEN = $\tau\, t_\gamma$ where t_γ is the number of generations when the termination condition with γ of C_{av} or C_{max} is satisfied and $\tau > 1$. To investigate the performance of H_ε gate, Schwefel function [12][13] is considered. Table 4.2 shows the effects of changing ε for the H_ε gate. As shown in the table, the results for $\varepsilon = 0.01$ were the best for the Schwefel function, although the average number of generations was larger than other results. It should be noted that if ε is too big, the performance would be worse than that of QiEA with the rotation gate ($\varepsilon = 0$). While a large ε (= 0.03) induces a fast premature convergence, a properly selected-small value of ε (= 0.01) provides better solutions. In particular, H_ε gate is recommended for a class of numerical optimization problems which have many local optima. We have employed other benchmark test, comparing results of our QiEA with differential evolution and particle swarm with 1000 function evaluations of several numerical problems with a suitable quantization of starting functions as an alternative to binary algorithms. In table 2 and 3 are shown results for these last benchmark tests, it is evident that QiEA was able to reach better results with much less evaluations than others two algorithms. Also, the algorithm was able to find better results than other two with the same number of function evaluations.

Table 1. Experimental results of the Schwefel function to show the effects of changing ε for H_ε gate. The population size was 15, the global migration period 100, the local group size 3, the number of observations 3, and the number of runs 30. γ of (4.4) was set to 0.9999. *b.*, *m.*, and *w.* mean *best*, *mean*, and *worst*, respectively. σ and t represent the standard deviation and the average number of generations, respectively.

	ε	0	0.005	0.01	0.015	0.02	0.03
f	b	1766.5	3.8×10^{-4}	3.8×10^{-4}	2.4×10^{-3}	0.2978	470.8
	m	2326.5	30.6	4.5×10^{-4}	8.4721	64.840	1041.6
	w	3462.1	420.9	6.7×10^{-4}	131.4	574.21	1875.1
	σ	550.8	69.65	6.1×10^{-5}	27.142	99.78	356.84
	t	5467.6	7001.8	7985.3	6874.0	5999.41	2891.3

Table 2. Mean number of function evaluations for each experiment

	QiEA	Diff Evolution	PSO
f1	40000	40000	300000
f2	19500	20000	250000
f3	65000	100000	80000
f4	10000	100000	80000

Table 3. Comparative results between QiEA, differential evolution and PSO with 1000 function evaluations

	QiEA	Diff Evolution	PSO
f1	8*10-19	3*10-17	21000
f2	5*10-12	2*10-9	nan
f3	0	2.3*10-21	0.678
f4	1.5*10-13	4*10-13	nan

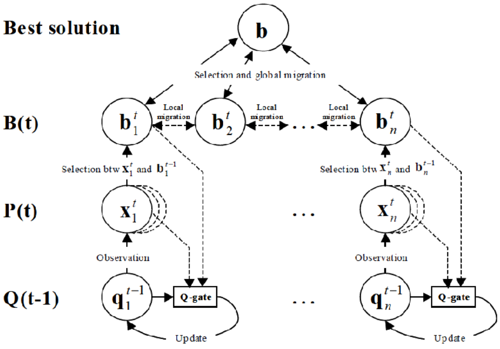


Fig. 2. Overall structure of QiEA

6 Conclusions

This paper presented a new quantum-inspired evolutionary algorithm with real representation that is better suited for numerical optimization problems than using binary representation. The new algorithm has been evaluated in several benchmark problems and showed very promising preliminary results, with better performance than other well-established algorithms. Further tests are needed in order to evaluate its robustness with other kinds of problems and a rigorous statistical analysis calculating e.g. standard deviation and average error is needed. Future works will include the use of the algorithm to optimize new benchmark functions and the use of the algorithm neuro-inspired [14] and other practical numerical optimization problems. Authors are working in these directions.

References

1. Deutsch, D.: Quantum Theory, the Church-Turing principle and the universal quantum computer. Pro.of the Royal Society of London A 400, 97–117 (1985)
2. Benioff, P.: The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. Journal of Statistical Physics 22, 563–591 (1980)
3. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Pro.of the Royal Society of London A 439, 553–558 (1992)
4. Simon, D.R.: On the Power of Quantum Computation. In: Proc. of the 35th Annual Symposium on Foundations of Computer Science, pp. 116–123. IEEE Press, Piscataway (1994)
5. Shor, P.W.: Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In: Proc. of the 35th Annual Symposium on Foundations of Computer Science, pp. 124–134. IEEE Press, Piscataway (1994)
6. Grover, L.K.: Quantum Mechanical Searching. In: Proc. of the 1999 Congress on Evolutionary Computation, vol. 3, pp. 2255–2261. IEEE Press, Piscataway (1999)
7. Lee, S.-C.: Quantum Computation. Technical report, Department of Physics, Korea Advanced Institute of Science and Technology, Korea
8. Kasabov, N.: Evolving Connectionist Systems: The Knowledge Engineering Approach, 2nd edn. Springer, London (2007)
9. Eshelman, L.J.: Genetic Algorithms. In: Back, T., Fogel, D.B., Michalewicz, Z. (eds.) Handbook of Evolutionary Computation, pp. B1.2:1–B1.2:11. OUP, New York (1997)
10. Porto, V.W.: Evolutionary Programming. In: Back, T., Fogel, D.B., Michalewicz, Z. (eds.) Handbook of Evolutionary Computation, pp. B1.4:1–B1.4:10. OUP, New York (1997)
11. Han, K.-H., Kim, J.-H.: Quantum-inspired Evolutionary Algorithm for a Class of Combinatorial Optimization. IEEE Trans. on Evolutionary Computation 6(6), 580–593 (2002)
12. Hirvensalo, M.: Quantum computing. Springer, Heidelberg (2004)
13. Han, K.-H.: andKim, J.-H.: Analysis of Quantum-inspired Evolutionary Algorithm. In: Proc. of the 2001 Int. Conf. on Artificial Intelligence, vol. 2, pp. 727–730. CSREA Press (2001)
14. Defoin-Platel, M., Schliebs, S., Kasabov, N.: Quantum-Inspired Evolutionary Algorithm: A Multimodel EDA. IEEE Transactions on Evolutionary Computation 13(6), 1218–1232 (2009)

Are You a Social Conformer?

Priyanka Garg, Irwin King, and Michael R. Lyu

The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
{priyanka,king,lyu}@cse.cuhk.edu.hk

Abstract. Social recommendations have been found to increase the product adoption probability. However, very few studies have considered the impact of social opinions on the users' evaluation of the product. In social networks, many times users' opinions are not completely independent from their friends and users tend to change their rating such that they are more similar to the social opinions. Understanding this behavior is important for developing accurate recommendation systems, precise information flow models and to launch effective viral marketing campaigns. In order to understand this phenomenon, we propose a novel formulation for the users ratings where every expressed rating is considered as a function of the social opinion along with the user preference and item characteristics. The proposed method helps in improving the prediction accuracy of users' rating by more than 2% in presence of social influence. Additionally, the learned model parameters reveal the degree of conformity of users. Detailed analysis of user social conformity show that more than 76% of users tend to conform to their friends to some extent. On an average, user ratings become more positive in presence of the social influence.

Keywords: Pattern Mining, Social Conformers, Recommender System.

1 Introduction

Social networks play a fundamental role in spreading information, ideas and technologies among their members. Often the decision to adopt a product is influenced by one's social connections. For example, positive friends reviews about a book encourages us to read it. Numerous studies have indicated that social recommendations result in an increase in the sales volume [2]. As a result, a large amount of research efforts have been devoted to understand the intricacies involved [5] and coming up with interesting applications like viral marketing [5], personalized recommender systems [6], etc.

However, most of the existing models have largely ignored the effect of social opinions on the *posterior* users evaluation of products i.e. the opinion the user form after experiencing the product. They either assume that the expressed opinion is same as the influencing opinion [5] or they are assumed to depend strictly on the product quality [1]. However many times, user's evaluation of the product, is not completely independent of her social circle and she tends to conform with social opinions. For example, a user reads a book and does not

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 694–701, 2012.

like it much. However lots of friends praise it and call it a really insightful book or “5/5”, then this might change the user’s opinion slightly and user might rate the book as “3”. Had she not interacted with her friend, she might have given a rating of “2”. This behavior usually arise because of the presence of social pressure and the innate difficulty involved in providing an absolute numerical rating to a product [8]. In such cases, social opinions can act as a reference rating and calibrates the user ratings such that they are not very different from the prevalent social opinion. We call this behavior as **social conformity** and the users who changes their rating as *social conformers*. Recently, this effect has been shown to exist on Goodreads and Douban [4].

Quantifying this behavior is important not just from the point of curiosity, but it is also crucial in improving the accuracy of personalized recommender systems and in developing better information flow models. The recommendation systems can boost the quality of recommendation by removing the social conformity bias, thus making the recommendation better tailored to users’ preference. While the information flow models can more accurately predict the further information cascade by accurately predicting the users’ opinions. However, it is a very **difficult task** to quantify the social conformity as for a given user and product we never get to know the two ratings, one under the social influence and one without it. All that is known is a single opinion expressed by the user. Thus, the key challenge is to identify what component of any rating corresponds to the user’s preference and what component corresponds to the social conformity.

In this paper, we account for social conformity and propose a novel formulation for the user’s ratings. Contrary to homophily based recommender systems [6], which try to learn user preference based on her friends’ preference, we focuse on the change of ratings at item level *caused* by the social influence. The proposed formulation represents every user rating as a function of social conformity and social opinion along with user’s preference and item’s characteristics. The social conformity down-weighs the user’s preference such that as the number of influential friends increases, the user’s rating become more similar to the social opinion. Further, the model parameters provide an intuitive interpretation of the social conformity behavior which reflect the degree a user conforms to her friend. It is important to note that different from the homophily based recommendation systems, we focus on the change of ratings at item level. Using this model, we explore the presence of social conformity on a real large scale dataset, Goodreads[1].

The key contributions of this paper are following.

1. We propose a novel formulation for user ratings that explicitly considers the social conformity. The proposed model improves the prediction accuracy of users’ ratings by more than 2% in presence of social influence.
2. The learned social conformity parameters are also verified by qualitatively comparing the discovered most influential users with the authoritative and most socially active users.
3. Based on the learned users’ degree of conformity, we find various interesting patterns on Goodreads that underline the impact of social conformity.

[1] http://www.goodreads.com/

2 Conformity Rating Model (CRM)

Notations. Let $G = (V, E)$ be a directed graph where every node $u \in V$ corresponds to a user in the social network and edge $(u, f) \in E$ exists if node f is a friend of node u. The user ratings for the set of items I, are stored in user-item matrix R, such that every element $r_{u,i}$ represents the rating for item i given by user u. Let the set of *active neighbors* who have posted their ratings for item i before user u be $A(u, i)$.

Problem Definition. The task is to predict the rating $r_{u,i}$ for item i given by user u, given the user-item matrix R and the set of active neighbors $A(u, i)$.

Social opinions calibrate user's inner rating $r^0_{u,i}$ such that they are not very different from them. To account for such social behavior, we propose the following social conformity based rating model CRM as

$$\hat{r}_{u,i} = r^0_{u,i} + conf \cdot (social\_opinion - r^0_{u,i}) \tag{1}$$

$$= \Big(1 - conf\Big) r^0_{u,i} + conf \cdot social\_opinion, \tag{2}$$

where $conf$ represents the degree by which a user conforms to the social opinion and $social\_opinion$ is the social opinion about the item i before the user u rates it. The rewritten form in Eq. (2) can also be seen as down-weighing the user's personal preference and giving higher weight to the friends' opinions. That is, if the user u has extremely high degree of social conformity then user u will change her rating such that it becomes same as the social opinion. Now we define each of the quantity $conf$, $social\_opinion$ and $r^0_{u,i}$ one by one.

- **User's Conformity** $conf$**.** We expect the degree of conformity $conf$ to take large values as the number of friends who have already rated the item increases. This phenomenon is known as the **bandwagon effect** in social sciences [3]. According to the bandwagon effect, as the number of individuals who believe in something increases, others tend to disregard their own opinions and also "hop on the bandwagon". That is, the social conformity is directly proportional to the number of friends with similar opinions. Thus, $conf = |A_{u,i}|$, because only active friends can affect the user's rating for the item. However, one can expect that users do not conform to all their friends equally. The friends who are regarded highly in the user's eyes, tend to affect their rating more. Hence, we introduce a parameter $\eta_{f,u}$ corresponding to every user and her friend pair. This parameter defines the degree by which user u conforms to the rating of its friend f. As the number of friends with high $\eta_{f,u}$ increases, the $conf$ can be expected to increase. Thus, we write

$$conf = \sum_{f \in A_{u,i}} \eta_{f,u}. \tag{3}$$

Since $conf$ can take maximum value of 1, we constraint $\eta_{f,u}$ such that $\sum_f \eta_{f,u} \leq 1$. Such linear forms of social influence have also been used in Linear threshold model [5] where the adoption probability of a product depends linearly on the active friends' influence.

- **Social Opinion** *social_opinion***.** We write the *social_opinion* as the sum of friends' opinions weighted according to $\eta_{f,u}$. This is because the opinions of friends with high $\eta_{f,u}$ affect the user's rating by the most amount. Thus, we have

$$social\_opinion = \frac{\sum_{f \in A_{u,i}} \eta_{f,u} \cdot r_{f,i}}{\sum_{f \in A_{u,i}} \eta_{f,u}}. \tag{4}$$

- **User's Inner Rating $r^0_{u,i}$.** To represent the user's inner rating $r^0_{u,i}$, we user one of the state of art recommendation models, Probability Matrix Factorization (PMF) method [7]. PMF model uses a small number of factors to represent the preference of users and item characteristics. The preference of users $q_u \in R^K$ and item characteristics $p_i \in R^K$ are represented by low dimensional vectors in latent space of dimensionality K. Then every rating is written as

$$\hat{r}^0_{u,i} = \mu + b_i + b_u + q_u^T \cdot p_i, \tag{5}$$

where μ is average user-item rating, b_i is item bias and b_u is user bias.

Thus, we finally have

$$\hat{r}_{u,i} = \Big(1 - \sum_{f \in A_{u,i}} \eta_{f,u}\Big)(\mu + b_i + b_u + q_u^T \cdot p_i) + \sum_{f \in A_{u,i}} \eta_{f,u} \cdot r_{f,i}.$$

Parameter Estimation. To estimate the model parameters b_i, b_u, q_u, p_i, $\eta_{f,u}$, we construct the objective function such that it minimizes the square of difference between observed user rating $r_{u,i}$ and estimated rating $\hat{r}_{u,i}$. Additionally, all parameters are regularized to avoid over fitting on the train dataset. Thus, our objective function is

$$\min \sum_{u,i} (r_{u,i} - \hat{r}_{u,i})^2 + \lambda_1 \Bigg(\sum_u b_u^2 + \sum_i b_i^2 + \sum_u ||q_u||^2 + \sum_i ||p_i||^2 \Bigg) + \lambda_2 \sum_{u,f} \eta^2_{f,u}$$

$$\text{s.t. } \eta_{f,u} \geq 0 \ \forall u, f; \ \sum_f \eta_{f,u} \leq 1 \ \forall u,$$

where λ_1 and λ_2 are the hyper-parameters which control the amount of regularization. The objective function is minimized by using the alternating minimization. In every first alternating step, we minimize the function with respect to the PMF model parameters b_i, b_u, q_u and p_i, using the steepest gradient decent method. Then in the second alternating step, we minimize the function with respect to $\eta_{f,u}$. Given the estimate $\hat{r}^0_{u,i}$ from first step, the objective function in the latter step can be written as the sum of small subproblem, each corresponding

Table 1. Goodreads data statistics

Users	Edges	Items	Ratings	Number of Authors
55,654	1,757,568	120,703	9,462,016	5,078

to one user. Since the set of parameters $\eta_{f,u}$ of every subproblem are different from the others, the objective function can be minimized by minimizing each of the sub problems separately. Thus, each of the sub problem can be minimized efficiently in parallel, using the gradient descent method.

3 Empirical Evaluation

We evaluate the effectiveness of CRM, both in terms of its ability to predict user ratings and its ability to identify the social influencers.

3.1 Goodreads Dataset

Goodreads is an online social books cataloging website, which permits users to rate books on $0-5$ scale (with 5 being the best) and share their reviews with friends. We use the dataset crawled by authors in [4]. The items and users are filtered such that every item has at least 10 ratings and every user has rated at least 5 books rated on 5 different dates and have at least 10 friends. This is to make sure the selected users are active users. In addition, we also crawl the profile pages of all the selected users. Users who have also authored books are marked as the authors. The statistics of the data is summarized in Table 1.

3.2 Prediction Accuracy

We evalute the ability of CRM to predict the users' ratings and compare its accuracy with the PMF method. The model parameters of both the PMF and the CRM, are trained on a train set and their performance is calulated on a test set. The train and test sets are constructed by splitting the user-item ratings in 4:1 ratio.

Performance Measure. The Root Mean Square Error (RMSE) metric is used for measuring the prediction accuracy. It is defined as $\sqrt{\frac{\sum_{u,i}(r_{u,i}-\hat{r}_{u,i})^2}{N}}$, where N is the number of ratings in the test set.

Observations. The RMSE values obtained using the CRM and PMF when $\lambda_1 = 1, \lambda_2 = 0.1$ are presented in Table 2. We can observe following.

- RMSE improves by more than 0.3% when social conformity is taken into consideration. Further, if we calculate RMSE value only for ratings who are potentially affected by the social influence ($conf > 0.1$), the RMSE improves by **more than** 2%.

Table 2. RMSE when $\lambda_1 = 1, \lambda_2 = 0.1$

		Number of test ratings	**PMF**	**CRM**
$K = 10$	All ratings	1,789,663	0.8556	**0.8520**
	Ratings with *conf* > 0.1	208,852	0.8476	**0.8254**
$K = 5$	All ratings	1,789,663	0.8472	**0.8441**
	Ratings with *conf* > 0.1	196,855	0.8471	**0.8280**

- **Sensitivity to K.** The RMSE value increases for both CRM and PMF model when K increase from 5 to 10. However, the drop in RMSE value is larger for PMF model than for CRM. This might be because a large value of latent space dimensionality K, can lead to over fitting on the training set. However, smaller impact on CRM underlines its robust performance.
- **Sensitivity to λ_2.** Effect of λ_2 (hyper-parameter to control the regularization) on the RMSE values is as per the expectations and is shown in Figure 1. The performance gets hurt if λ_2 is too large (≥ 1) or when it is too small (≤ 0.01). Higher value of λ_2 forces the selection of small social conformity factors $\eta_{f,u}$ and thereby under fits the model. While very small value leads to over-fitting on the training set. The best RMSE value is achieved when $\lambda_2 = 0.1$, though performance is reasonably robust around this value.

3.3 Influencers Quality

We evaluate the quality of the learned $\eta_{f,u}$ parameters by analyzing the properties of most influential users. In our setting, the users who have maximum effect on their friends' ratings are the social influencers. Formally, we define the social influence of a user u as $\sum_f \eta_{u,f}$ (Note that it is defined by reversing the conformity $\eta_{f,u}$ direction). We expect that the top influencers should have higher authority and higher number of social connections. Hence, we rank the users based on their social influence and study their degree and authority.

- **Average degree of top influential users.** The average degree (number of friends) of top x most influential users as function of x is plotted in Figure 2. It can be noticed that top 200 influential users have the highest average degree and as we consider more and more top users as the influential ones, their average degree starts to fall.
- **Authority of top influential users.** We validate the authority of the influential users by checking if they have also authored the books. This is a reasonable criterion as the book authors have higher perceived authority among their friends. The plot of percentage of authors among the top x influential users is shown in Figure 3. It can be noted that percentage of authors is very high among the top influencers. More than 45% authors appear in the top 200 influencers and there are only 12% authors among the top 5000 users, while the entire dataset has approximately 9% authors. Thus, as we keep widening the value of x, the ratio of authors to non-authors approaches to the ratio of entire dataset.

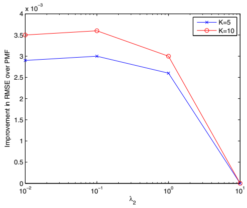


Fig. 1. Improvement over PMF as $\lambda2$ varies

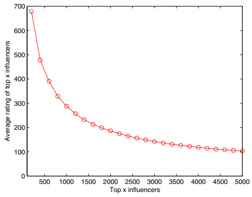


Fig. 2. Average degree of top influencer

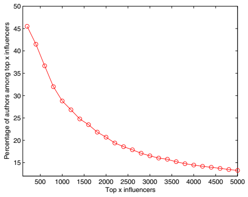


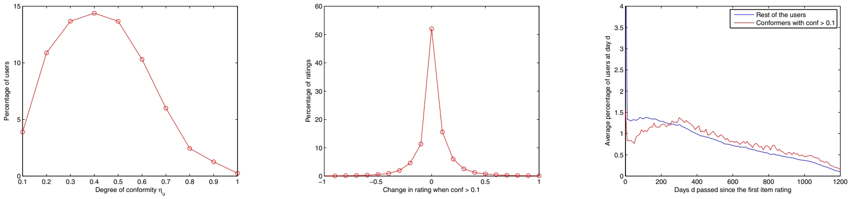
Fig. 3. Authors among top influencer

Both the observations show that CRM is able to **identify the social influencers** accurately.

4 Social Conformity Analysis

Having verified CRM model, in this section we explore the nature of social conformity. We seek to answer following the questions.

- How many users conform to their friends?
  The distribution of user conformity $\eta_u = \sum_f \eta_{f,u}$ is presented in Figure 4(a). It can be noticed that more than 76% users have $\eta_u > 0$. Among these users, most of them belong to the 0.2 to 0.6 interval. That is, most of the users in the network display some sort of conformity to their friends. In gerernal, we find that most of the users conform to only one friend and less than 9% users conform to more than 14 friends.
- By how much amount the user ratings change because of social opinions?
  We plot the distribution of change in ratings caused by the social opinions for ratings with $conf > 0.1$. Results are presented in Figure 4(b). It can be noted that, most of the ratings change is between -0.5 and 0.5. Additionally, it is interesting that more ratings change by positive factor than by the negative factor. 15% rating changes by +0.1 amount while only 12% ratings changes to -0.1 amounts.
- When does the conformity prevail the most?
  For each item, we find the percentage of social ratings with $conf > 0.1$ that appeared in between day d and day $d + 10$ since first rating is posted. Then, we calculate their average over all the items and plot them against day d. Similarly we plot the ratings with $conf \leq 0.1$. Results are presented in Figure 4(c). It can be seen that two kinds of ratings follow different pattern. The ratings with $conf \leq 0.1$ have maximum presence during the start of the information cascade and their percentage decays slowly as the time passes by. While the ratings with $conf > 0.1$ have relatively smaller presence at the start. As the time passes by their percentage increases and peaks at around 300 days passed. After that, their percentage falls with time and follows similar pattern as the other ratings. In general, we find that users with higher value of η_u tend to post their ratings late.



(a) Distribution of user conformity (b) Change in ratings caused by social opinions (c) % of conformers as the item cascade unfolds

Fig. 4. Patterns of social conformity

5 Conclusion

We propose a novel formulation for the user ratings CRM that explicitly considers the social opinions. The CRM is shown to be effective in both improving the prediction accuracy of user's rating and in accurately identifying the social influencers. Further, several interesting patterns have emerged. We find that more than 76% of users show some degree of conformity with their friends. To our surprise, our friends opinion makes our posterior evaluation of the product more positive then negative, which is certainly a good news for the viral marketing strategy. Similar to the item product adopters, the users with high conformity tend to post their rating during the later part of the information cascade. We hope that the patterns found in this paper, would help in developing better recommendation systems and information propagation models.

References

1. Chen, W., Colin, A., Cumming, R., Ke, T., Liu, Z., Rincon, D., Sun, X., Wang, Y., Wei, W., Yuan, Y.: Influence maximization in social networks when negative opinion may emerge and propagate. In: SDM (2011)
2. Chevalier, J.A., Mayzlin, D.: The Effect of Word of Mouth on Sales: Online Book Reviews. Journal of Marketing Research 43(3), 345–354 (2005)
3. Gisser, M., McClure, J.E., Okten, G., Santoni, G.: Some anomalies arising from bandwagons that impart upward sloping segments to market demand. Econ. Journal Watch 6(1), 21–34 (2009)
4. Huang, J., Cheng, X.Q., Shen, H.W., Zhou, T., Jin, X.: Exploring social influence via posterior effect of word-of-mouth recommendations. In: WSDM (2012)
5. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: KDD (2003)
6. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: WSDM (2011)
7. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, vol. 20 (2008)
8. Hwang, S.W., Lee, M.W.: A uncertainty perspective on qualitative preference. In: UAI (2009)

Canonical Duality Theory and Algorithm for Solving Challenging Problems in Network Optimisation

Ning Ruan[1,2] and David Yang Gao[1]

[1] School of Sciences, Information Technology and Engineering,
University of Ballarat, Ballarat, VIC 3353, Australia
[2] Department of Mathematics and Statistics,
Curtin University, Perth, WA 6845, Australia
n.ruan@ballarat.edu.au

Abstract. This paper presents a canonical dual approach for solving a general nonconvex problem in network optimization. Three challenging problems, sensor network location, traveling salesman problem, and scheduling problem are listed to illustrate the applications of the proposed method. It is shown that by the canonical duality, these nonconvex and integer optimization problems are equivalent to unified concave maximization problem over a convex set and hence can be solved efficiently by existing optimization techniques.

Keywords: Global Optimization, Canonical Duality, Wireless Network, Traveling Salesman Problem, Scheduling Problem.

1 Introduction

Let us consider the following nonconvex (primal) optimization problem that arises in a wide range of applications:

$$(\mathcal{P}): \ \min\left\{P(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} - \mathbf{f}^T\mathbf{x} + W(\mathbf{x}) \ : \ \mathbf{x} \in \mathcal{X}_a\right\}, \tag{1}$$

where $Q = \{q_{ij}\} \in \mathbb{R}^{n\times n}$ is a given symmetric matrix, $\mathbf{f} \in \mathbb{R}^n$ is a given vector, $\mathcal{X}_a \subset \mathbb{R}^n$ is a convex open set, and $W(\mathbf{x})$ is a nonconvex function. Note that in the context of constrained optimization problems, the function $W(\mathbf{x})$ could be simply defined as a (nonsmooth) indicator function of a feasible space $\mathcal{X}_c$:

$$W(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \mathcal{X}_c \\ +\infty & \text{otherwise.} \end{cases} \tag{2}$$

If $\mathcal{X}_a = \mathbb{R}^n$ and $\mathcal{X}_c = \{\mathbf{x} \in \mathbb{R}^n |\ \mathbf{A}\mathbf{x} \le \mathbf{b}, \ \ \mathbf{l} \le \mathbf{x} \le \mathbf{u}\}$, where $\mathbf{A} \in \mathbb{R}^{m\times n}$ is a matrix, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$ are given vectors, then Problem $(\mathcal{P})$ reduces to a linearly constrained quadratic program:

$$(\mathcal{P}_q): \quad \min\left\{P(\mathbf{x}) = \frac{1}{2}\,\mathbf{x}^T Q\mathbf{x} - \mathbf{f}^T\mathbf{x} \ \ : \ \ \mathbf{A}\mathbf{x} \le \mathbf{b}, \ \mathbf{l} \le \mathbf{x} \le \mathbf{u}, \ \mathbf{x} \in \mathbb{R}^n\right\}. \tag{3}$$

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 702–709, 2012.

It is well-known that even this most simple problem is NP-hard if Q is indefinite and considerable efforts have been devoted to solve this type of problems.

The key idea of the canonical dual transformation is to choose a certain geometrically reasonable measure (operator) $\boldsymbol{\varepsilon} = \Lambda(\mathbf{x}) : \mathcal{X}_a \subset \mathbb{R}^n \to \mathcal{E}_a \subset \mathbb{R}^m$ such that the nonconvex functional $W(\mathbf{x})$ can be recast by adopting the canonical form $W(\mathbf{x}) = V(\Lambda(\mathbf{x}))$. Thus, the primal problem $(\mathcal{P})$ can be written in the following canonical form:

$$\min \{P(\mathbf{x}) = V(\Lambda(\mathbf{x})) - U(\mathbf{x}) \ : \ \mathbf{x} \in \mathcal{X}_a\}, \tag{4}$$

where $U(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{f}^T\mathbf{x}$. For the given canonical function $V(\boldsymbol{\varepsilon})$, its Legendre conjugate $V^*(\boldsymbol{\varsigma})$ can be defined uniquely by the Legendre transformation, and the following canonical duality relations hold:

$$\boldsymbol{\varsigma} = \nabla V(\boldsymbol{\varepsilon}) \ \Leftrightarrow \ \boldsymbol{\varepsilon} = \nabla V^*(\boldsymbol{\varsigma}) \ \Leftrightarrow \ V(\boldsymbol{\varepsilon}) + V^*(\boldsymbol{\varsigma}) = \boldsymbol{\varepsilon}^T\boldsymbol{\varsigma}. \tag{5}$$

In finite deformation mechanics, the one-to-one canonical duality relation $\boldsymbol{\varsigma} = \nabla V(\boldsymbol{\varepsilon})$ is called the canonical constitutive law [1]. By this canonical duality, the nonconvex term $W(\mathbf{x}) = V(\Lambda(\mathbf{x}))$ in the problem $(\mathcal{P})$ can be replaced by $\Lambda(\mathbf{x})^T\boldsymbol{\varsigma} - V^*(\boldsymbol{\varsigma})$ such that the nonconvex function $P(\mathbf{x})$ is reformulated as

$$\Xi(\mathbf{x}, \boldsymbol{\varsigma}) = \Lambda(\mathbf{x})^T\boldsymbol{\varsigma} - V^*(\boldsymbol{\varsigma}) - U(\mathbf{x}), \tag{6}$$

which is the so-called *total complementary function* introduced by Gao and Strang in nonconvex mechanics [1]. By using this total complementary function, the canonical dual function can be formulated as

$$P^d(\boldsymbol{\varsigma}) = \text{sta}\{\Xi(\mathbf{x}, \boldsymbol{\varsigma}) \ : \ \mathbf{x} \in \mathcal{X}_a \ \} = U^\Lambda(\boldsymbol{\varsigma}) - V^*(\boldsymbol{\varsigma}), \tag{7}$$

where $U^\Lambda(\boldsymbol{\varsigma}) = \text{sta}\{\Lambda(\mathbf{x})^T\boldsymbol{\varsigma} - U(\mathbf{x}) \ : \ \mathbf{x} \in \mathcal{X}_a\}$ is the so-called Λ-conjugate of U, which is defined on the dual feasible space $\mathcal{S}_a$.

2 Challenging Problems and Applications

2.1 Wireless Network Localization

Consider the following general nonlinear programming problem arising from Euclidean distance geometry (see [3]):

$$(\mathcal{P}) \quad \min \left\{ P(\mathbf{X}) = \sum_{(i,j)\in\mathcal{S}} \frac{1}{2} w_{ij} \left(\frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2 - \mu_{ij} \right)^2 \right. \\ \left. + \frac{1}{2}\langle \mathbf{X}, \mathbf{A}\mathbf{X}\rangle - \langle \mathbf{X}, \mathbf{T}\rangle \ | \mathbf{X} \in \mathcal{X}_a \right\},$$

where the decision variable $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] = \{x_i^\alpha\}_{i,\alpha} \in \mathbb{R}^{r\times n}$ is a matrix (two-point tensor) with each column $\mathbf{x}_i \in \mathbb{R}^r$ as a position of each sensor such that

$$\|\mathbf{x}_i - \mathbf{x}_j\| = \left(\sum_{\alpha=1}^{r} (x_i^\alpha - x_j^\alpha)^2 \right)^{\frac{1}{2}}$$

denotes the Euclidian distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, $(i,j) \in \mathcal{S} = \{1, 2, \cdots, n\}$; $\mathcal{X}_a \subset \mathbb{R}^{d\times n}$ is a feasible set; $\mathbf{T} = \{T^i_\alpha\} \in \mathcal{X}^* = \mathbb{R}^{n\times d}$ is a given matrix; $w_{ij} \geq 0$ and $\mu_{ij} \geq 0$ $(\forall i, j \in \mathcal{S})$ are given weights and parameters for each pair $(\mathbf{x}_i, \mathbf{x}_j)$, respectively; $\mathbf{A} = \{A^{i,\beta}_{\alpha,j}\}$ is a fourth-order symmetric tensor, and $\mathbf{AX} = \{\sum_{j=1}^n \sum_{\beta=1}^r \mathbf{A}^{i,\beta}_{\alpha,j} x^\beta_j\}_{i,\alpha}$, the bilinear form $\langle \mathbf{X}, \mathbf{T}\rangle : \mathcal{X}_a \times \mathcal{X}^* \to \mathbb{R}$ is defined as

$$\langle \mathbf{X}, \mathbf{T}\rangle = tr(\mathbf{XT}) = \sum_{i=1}^n \sum_{\alpha=1}^d X^\alpha_i T^i_\alpha.$$

Canonical Geometric Measure and Dual Problem Since

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n) \in \mathbb{R}^{r\times n},$$

we have the identity

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j) = (\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{X}^T \mathbf{X}(\mathbf{e}_i - \mathbf{e}_j),$$

where $\mathbf{e}_i$ is the i-th standard unit vector in $\mathbb{R}^n$. Introducing a linear (difference) operator $\mathbf{D} : \mathcal{X}_a \to \mathbb{R}^{r\times n\times n}$ such that

$$\mathbf{DX} = \{\mathbf{X}(\mathbf{e}_i - \mathbf{e}_j)\} = \{\mathbf{x}_i - \mathbf{x}_j\},$$

the *canonical strain measure* $\boldsymbol{\xi}$ can be defined as

$$\boldsymbol{\xi} = \{\xi_{ij}\} = \Lambda(\mathbf{X}) = \frac{1}{2}(\mathbf{DX})^T(\mathbf{DX}) = \frac{1}{2}\left\{(\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{X}^T \mathbf{X}(\mathbf{e}_i - \mathbf{e}_j)\right\},$$

where Λ is the so-called *geometrical nonlinear operator* from $\mathcal{X}_a \subset \mathbb{R}^{r\times n}$ into

$$\mathcal{V}_a = \{\boldsymbol{\varepsilon} \in \mathbb{R}^{n\times n} |\ \ \boldsymbol{\xi} = \boldsymbol{\xi}^T, \ \ \boldsymbol{\xi} \succeq 0, \ \xi_{ii} = 0, \ i = 1, \cdots, n\}.$$

Clearly, $\xi_{ij} = \frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2$, which is corresponding to the Cauchy-Riemann strain tensor in finite deformation theory. By introducing a quadratic function $V : \mathcal{V}_a \to \mathbb{R}$,

$$V(\boldsymbol{\xi}) = \frac{1}{2}\sum_{i,j} w_{ij}(\xi_{ij} - \mu_{ij})^2 = \frac{1}{2}\langle(\boldsymbol{\xi} - \boldsymbol{\mu}); \mathbf{W} \circ (\boldsymbol{\xi} - \boldsymbol{\mu})\rangle,$$

where $\mathbf{W} = \{w_{ij}\}$, $\boldsymbol{\mu} = \{\mu_{ij}\}$, $\mathbf{W} \circ \boldsymbol{\mu} = \{w_{ij}\mu_{ij}\}$ represents the Hadamard product of two matrices, and $\langle *; *\rangle$ denotes the bilinear operator of two matrices. The primal problem $(\mathcal{P})$ can now be reformulated in the canonical form:

$$(\mathcal{P}) : \ \ \min \left\{ \Pi(\mathbf{X}) = V(\Lambda(\mathbf{X})) + \frac{1}{2}\langle \mathbf{X}, \mathbf{AX}\rangle - \langle \mathbf{X}, \mathbf{T}\rangle : \ \ \mathbf{X} \in \mathcal{X}_a \right\}.$$

By the canonical dual transformation, the canonical dual problem can be proposed as follows:

$$(\mathcal{P}^d) : \ \ \text{sta}\left\{ P^d(\boldsymbol{\varsigma}) = -\frac{1}{2}\langle \mathbf{G}^+(\boldsymbol{\varsigma})\mathbf{T}, \mathbf{T}\rangle - \frac{1}{2}\langle \boldsymbol{\varsigma}; \mathbf{W}^{-1} \circ \boldsymbol{\varsigma}\rangle - \langle \boldsymbol{\mu}; \boldsymbol{\varsigma}\rangle \ \mid \ \boldsymbol{\varsigma} \in \mathcal{S}_a \right\},$$

where, $\mathbf{G}(\varsigma) = \mathbf{A} + \mathbf{D}^T\varsigma\mathbf{D}$ with $\mathbf{D}^T\varsigma = (\mathbf{e}_i^T - \mathbf{e}_j^T)\varsigma$, $\mathbf{G}^+$ represents the generalized inverse of $\mathbf{G}$, the dual feasible space $\mathcal{S}_a$ is a subset of $\mathbb{R}^{n\times n}$ such that for a given $\mathbf{T}$, the matrix equation $\mathbf{G}(\varsigma)\ \mathbf{X} = \mathbf{T}$ is solvable on $\mathcal{S}_a$.

Theorem 1 (Complementary-Dual Principle). *The problem* $(\mathcal{P}^d)$ *is a canonical dual of the primal problem* $(\mathcal{P})$ *in the sense that if* $\bar{\varsigma}$ *is a critical point of* $(\mathcal{P}^d)$*, then*

$$\bar{\mathbf{X}} = \mathbf{G}^+(\bar{\varsigma})\mathbf{T} \tag{8}$$

is a critical point of $(\mathcal{P})$ *and*

$$P(\bar{\mathbf{X}}) = P^d(\bar{\varsigma}).$$

In order to identify extremality of the analytical solution (8), we need to introduce a useful feasible space

$$\mathcal{S}_a^+ = \{\varsigma \in \mathcal{S}_a \mid \ \mathbf{G}(\varsigma) \succ 0\}.$$

Theorem 2. *Suppose that* $\bar{\varsigma} \in \mathcal{S}_a^+$ *is a critical point of the canonical dual function* $P^d(\bar{\varsigma})$ *and* $\bar{\mathbf{X}} = \mathbf{G}^+(\bar{\varsigma})\mathbf{T}$*. Then,* $\bar{\mathbf{X}}$ *is a global minimizer of* $P(\mathbf{X})$ *on* $\mathbb{R}^{r\times n}$ *if and only if* $\bar{\varsigma}$ *is a global maximizer of* $P^d(\varsigma)$ *on* $\mathcal{S}_a^+$*, i.e.,*

$$P(\bar{\mathbf{X}}) = \min_{\mathbf{X}\in\mathbb{R}^{r\times n}} P(\mathbf{X}) \Leftrightarrow \max_{\varsigma\in\mathcal{S}_a^+} P^d(\varsigma) = P^d(\bar{\varsigma}). \tag{9}$$

This theory shows that if the canonical dual function $P^d(\varsigma)$ has a critical point in $\mathcal{S}_a^+$, then the nonconvex primal problem $(\mathcal{P})$ is equivalent to a concave maximization problem $(\mathcal{P}^d)$ over a convex space $\mathcal{S}_a^+$, which can be solved easily by well-developed optimization methods.

2.2 Traveling Salesman Problem

Consider the well-known Traveling salesman problem (TSP), which we need to determine the shortest closed path passing through a set of n cities, with each city visited exactly once. Suppose $\mathcal{N} = \{1, 2, \cdots, n\}$ is the set of TSP cities, and the distance between city i and city j is given by d_{ij}. Assume

$$d_{ii} = 0,\ d_{ij} = d_{ji},\ \forall i, j \in \mathcal{N}.$$

Define a Boolean decision variable x_{ij} according to

$$x_{ij} = \begin{cases} 1 & \text{if city } i \text{ is in the } jth \text{ position,} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

To make sure the round trip, we assume

$$x_{i0} = x_{in},\ x_{i1} = x_{i(n+1)},\ \forall i, j \in \mathcal{N}.$$

Let $\mathbf{X} = \{x_{ij}\} \in \mathbb{R}^{n\times n}$, the Traveling salesman problem can be represented by following quadratic programming problem [8]:

$$
\begin{aligned}
(\mathcal{P}) \quad \text{Minimize } \; & P(\mathbf{X}) = \sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{n} x_{ij} d_{ik} \cdot (x_{k(j+1)} + x_{k(j-1)}) \\
\text{subject to } \; & \sum_{j=1}^{n} x_{ij} = 1, \ \forall i \in \mathcal{N}, \quad \sum_{i=1}^{n} x_{ij} = 1, \ \forall j \in \mathcal{N}, \\
& x_{ij} \in \{0,1\}, \ \forall i,j \in \mathcal{N}.
\end{aligned}
$$

Canonical Dual Problem. Let

$$G(\boldsymbol{\mu}) = \mathbf{A} + 2\text{Diag }(\boldsymbol{\mu}),$$

$$F(\boldsymbol{\sigma},\boldsymbol{\tau},\boldsymbol{\mu}) = \boldsymbol{\mu} - \mathbf{C}^T\boldsymbol{\sigma} - D^T\boldsymbol{\tau}.$$

By the canonical dual transformation [1], the canonical dual problem can be stated as follows:

$$
\begin{aligned}
(\mathcal{P}^d) \quad \text{Maximize } \; & P^d(\boldsymbol{\sigma},\boldsymbol{\tau},\boldsymbol{\mu}) = -\frac{1}{2}F(\boldsymbol{\sigma},\boldsymbol{\tau},\boldsymbol{\mu})^T G^{\dagger}(\boldsymbol{\mu}) F(\boldsymbol{\sigma},\boldsymbol{\tau},\boldsymbol{\mu}) - \boldsymbol{\sigma}^T\mathbf{e} - \boldsymbol{\tau}^T\mathbf{e} \\
\text{subject to } \; & \boldsymbol{\sigma} \neq 0, \ \boldsymbol{\tau} \neq 0, \boldsymbol{\mu} \neq 0, \\
& \boldsymbol{\sigma} \in \mathbb{R}^n, \boldsymbol{\tau} \in \mathbb{R}^n, \boldsymbol{\mu} \in \mathbb{R}^{nn},
\end{aligned}
$$

where, $\mathbf{A} = \{a_{st}\} \in \mathbb{R}^{nn\times nn}$ is a block matrix, which satisfies

$$
a_{st} = \begin{cases}
d_{ik}, & \text{if } s = (i-1)N + j \text{ and } t = (k-1)N + (j-1), \ \forall i,k,j \in \mathcal{N}, \\
d_{ik}, & \text{if } s = (i-1)N + j \text{ and } t = (k-1)N + (j+1), \ \forall i,k,j \in \mathcal{N}, \\
d_{ki}, & \text{if } s = (k-1)N + (j-1) \text{ and } t = (i-1)N + j, \ \forall i,k,j \in \mathcal{N}, \\
d_{ki}, & \text{if } s = (k-1)N + (j+1) \text{ and } t = (i-1)N + j, \ \forall i,k,j \in \mathcal{N}, \\
0, & \text{otherwise},
\end{cases}
$$

$$
\mathbf{C} = \begin{bmatrix}
1 \cdots 1 \; 0 \cdots 0 \cdots 0 \cdots 0 \\
0 \cdots 0 \; 1 \cdots 1 \cdots 0 \cdots 0 \\
\vdots \ddots \vdots \; \vdots \ddots \vdots \ddots \vdots \ddots \vdots \\
0 \cdots 0 \; 0 \cdots 0 \cdots 1 \cdots 1
\end{bmatrix} \in \mathbb{R}^{n\times nn},
$$

$$
D = \begin{bmatrix}
1 \; 0 \cdots 0 \cdots\cdots 1 \; 0 \cdots 0 \\
0 \; 1 \cdots 0 \cdots\cdots 0 \; 1 \cdots 0 \\
\vdots \; \vdots \ddots \vdots \;\; \vdots \;\; \vdots \;\; \vdots \; \vdots \ddots \vdots \\
0 \; 0 \cdots 1 \cdots\cdots 0 \; 0 \cdots 1
\end{bmatrix} \in \mathbb{R}^{n\times nn},
$$

$$\mathbf{e} = [1,\cdots,1,\cdots,1,\cdots,1]^T \in \mathbb{R}^n.$$

Theorem 3 (Complementary-Dual Principle). *Problem $(\mathcal{P}^d)$ is a canonical dual of Problem $(\mathcal{P})$ in the sense that if $(\bar{\boldsymbol{\sigma}}, \bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}})$ is a KKT solution of Problem $(\mathcal{P}^d)$, then the vector $\bar{\mathbf{X}} = \{x_{ij}\} \in \mathbb{R}^{n\times n}$ defined by*

$$x_{ij} = y_{(i-1)n+j},\ \forall i,j \in \mathcal{N},\ \textit{and}\ \bar{\mathbf{y}} = G^{\dagger}(\bar{\boldsymbol{\mu}})F(\bar{\boldsymbol{\sigma}}, \bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}}) \in \mathbb{R}^{nn} \qquad (11)$$

is a KKT solution of Problem $(\mathcal{P})$ and $P(\bar{\mathbf{X}}) = P^d(\bar{\boldsymbol{\sigma}}, \bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}})$.

To continue, let the feasible space $\mathcal{X}$ of problem $(\mathcal{P})$ and the dual feasible space $\mathcal{Z}$ be defined by

$$\begin{aligned} \mathcal{X} &= \left\{ \mathbf{X} \in \mathbb{R}^{n\times n} : \sum_{j=1}^{n} x_{ij} = 1,\ \sum_{i=1}^{n} x_{ij} = 1,\ x_{ij} \in \{0,1\},\ \forall i,j \in \mathcal{N} \right\} \\ \mathcal{Z} &= \{(\boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\mu}) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{nn} : \boldsymbol{\sigma} \neq 0,\ \boldsymbol{\tau} \neq 0, \boldsymbol{\mu} \neq 0\}, \\ \mathcal{Z}_a^+ &= \{(\boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\mu}) \in \mathcal{Z} : G(\boldsymbol{\mu}) \succ 0\}. \end{aligned}$$

We have the following theorem.

Theorem 4. *Assume that $(\bar{\boldsymbol{\sigma}}, \bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}})$ is a KKT point of $P^d(\boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\mu})$ and $\bar{X}$ defined by (11). If $(\bar{\boldsymbol{\sigma}}, \bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}}) \in \mathcal{Z}_a^+$, then $\bar{X}$ is a global minimizer of $P(\mathbf{X})$ and $(\bar{\boldsymbol{\sigma}}, \bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}})$ is a global maximizer of $P^d(\boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\mu})$ with*

$$P(\bar{X}) = \min_{\mathbf{X}\in\mathcal{X}} P(\mathbf{X}) = \max_{(\boldsymbol{\sigma},\boldsymbol{\tau},\boldsymbol{\mu})\in\mathcal{Z}_a{}^+} P^d(\boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\mu}) = P^d(\bar{\boldsymbol{\sigma}}, \bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}}) \qquad (12)$$

2.3 Scheduling Problem in Supply Chain

In project scheduling, a set of resource-constrained jobs has to be scheduled so as to minimize a given objective resources. The scheduling problem has a variety of applications in manufacturing, production planning, project management, and elsewhere.

We consider the problem to minimize the total cost of a schedule when the jobs are subject to temporal constraints only (i.e., there are no resource constraints). A common way to model scheduling problems as integer linear programs is to use time indexed variables. Let

$$x_{jt} = \begin{cases} 1 \text{ if job } j \text{ starts at time } t, \\ 0 \ \text{ otherwise,} \end{cases}$$

where, $j \in J = 0, \cdots, n$. Jobs 0 and n are assumed to be artificial jobs indicating the project start and the project completion, respectively, d_{ij} be the integral length of a time lag (i,j) between two jobs $i,j \in J$, and let $L \subseteq J \times J$ be the set of all given time lags, T be the deadline of the project, and $t = 0, \cdots, T$, p_i be the processing time of activity i, the precedence relation $(i,j) \in L$ if activity j cannot start before activity i completes. Finally, let w_{jt} be the net present value of activity j when starting at time t. This leads to the following integer linear program:

$$(\mathcal{P}) \quad \text{Minimize } \; P(\mathbf{x}) = \sum_{j=0}^{n} \sum_{t=0}^{T} w_{jt} x_{jt} \tag{13}$$

$$\text{subject to } \; \sum_{t=0}^{T} x_{jt} = 1, \; j \in J, \;\; \sum_{t=0}^{T} t(x_{jt} - x_{it}) \geq d_{ij}, \; (i,j) \in L, \tag{14}$$

$$x_{jt} \in \{0,1\}, \; j \in J, \; t = 0, \cdots, T. \tag{15}$$

Canonical Dual Problem. Let

$$\begin{aligned} \mathbf{X} &= [x_{00}, \cdots, x_{0T}, \cdots, x_{n0}, \cdots, x_{nT}]^T, \\ \mathbf{W} &= [w_{00}, \cdots, w_{0T}, \cdots, w_{n0}, \cdots, w_{nT}]^T, \\ \mathbf{D} &= [d_{00}, \cdots, d_{0n}, \cdots, d_{n0}, \cdots, d_{nn}]^T, \; d_{ij} = 0 \text{ if } i \geq j \end{aligned}$$

and

$$\mathbf{B} = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 1 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{(n+1)\times[(T+1)\times(n+1)]},$$

$$\mathbf{A} = \begin{bmatrix} 0 & \cdots & T & 0 & \cdots & -T & \cdots\cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots\ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & T & 0 & \cdots & 0 & \cdots\cdots & 0 & \cdots & 0 & 0 & \cdots & -T \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots\ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots\cdots & 0 & \cdots & T & 0 & \cdots & -T \end{bmatrix} \in \mathbb{R}^{[(n+1)\times(n+1)]\times[(T+1)\times(n+1)]},$$

By the canonical dual theory [1], the canonical dual problem can be stated as follows:

$$\begin{aligned} (\mathcal{P}^d) \quad & \text{Maximize } \; P^d(\boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\mu}) = -\frac{1}{2} F(\boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\mu})^T \mathbf{G}^+(\boldsymbol{\mu}) F(\boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\mu}) - \boldsymbol{\sigma}^T \mathbf{e} + \boldsymbol{\tau}^T \mathbf{e} \\ & \text{subject to } \; \boldsymbol{\sigma} > 0, \; \boldsymbol{\tau} \geq 0, \boldsymbol{\mu} > 0, \\ & \qquad\qquad \boldsymbol{\sigma} \in \mathbb{R}^{n+1}, \boldsymbol{\tau} \in \mathbb{R}^{(n+1)\times(n+1)}, \boldsymbol{\mu} \in \mathbb{R}^{(T+1)\times(n+1)}, \end{aligned}$$

where,

$$\mathbf{G}(\boldsymbol{\mu}) = 2\text{Diag }(\boldsymbol{\mu}), \; F(\boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\mu}) = \boldsymbol{\mu} - \mathbf{W} - \mathbf{B}^T \boldsymbol{\sigma} - \mathbf{A}^T \boldsymbol{\tau}.$$

And we have complementary-dual principle and optimization criterion similar to Theorem 3 and Theorem 4.

3 Conclusions

We have presented simple applications of the canonical duality theory for three challenging problems. A general analytical solution is obtained by the complementary-dual principle. Results show that by using the canonical dual transformation, the nonconvex primal problem and integer programming problem can be converted to a unified concave maximization dual problem, which can be solved by well-developed convex minimization techniques. The idea and the method presented in this article can be used and generalized to solve much more difficult problems in global optimization, network communication, and scientific computations (see [2, 4–7]). The development of techniques is essential to extrapolate the complexities of the real world.

Acknowledgments. This paper was partially supported by a grant (AFOSR FA9550-10-1-0487) from the US Air Force Office of Scientific Research. Dr. Ning Ruan was supported by a funding from the Australian Government under the Collaborative Research Networks (CRN) program.

References

1. Gao, D.Y.: Duality Principles in Nonconvex Systems: Theory, Methods and Applications. Kluwer Academic Publishers, Dordrecht (2000)
2. Gao, D.Y., Ruan, N.: Solutions to Quadratic Minimization Problems with Box and Integer Constraints. J. Global Optim. 47, 463–484 (2010)
3. Gao, D.Y., Ruan, N., Pardalos, P.M.: Canonical Dual Solutions to Sum of Fourth-Order Polynomials Minimization Problems with Applications to Sensor Network Localization. In: Boginski, V.L., Commonder, C.W., Pardalos, P.M., Ye, Y.Y. (eds.) Sensors: Theory, Algorithms and Applications, vol. 61, pp. 37–54. Springer (2012)
4. Gao, D.Y., Ruan, N., Sherali, H.D.: Solutions and Optimality Criteria for Nonconvex Constrained Global Optimization Problems. J. Global Optim. 45, 473–497 (2009)
5. Gao, D.Y., Watson, L.T., Easterling, D.R., Thacker, W.I., Billups, S.C.: Solving the Canonical Dual of Box- and Integer-Constrained Nonconvex Quadratic Programs via a Deterministic Direct Search Algorithm. Optim. Method Softw. (2011), doi:10.1080/10556788.2011.641125
6. Gao, D.Y., Wu, C.Z.: On the Triality Theory for a Quartic Polynomial Optimization Problem. J. Ind. Manag. Optim. 8, 229–242 (2012)
7. Ruan, N., Gao, D.C., Jiao, Y.: Canonical Dual Least Square Method for Solving General Nonlinear Systems of Equations. Comput. Optim. Appl. 47, 335–347 (2010)
8. Smith, K.: An Argument for Abandoning the Traveling Salesman Problem as a Neural-Network Benchmark. IEEE Trans. Neural Networks 7, 1542–1544 (1996)

Author Index