LNCS 5864

Chi Sing Leung
Minho Lee
Jonathan H. Chan (Eds.)

# Neural Information Processing

16th International Conference, ICONIP 2009
Bangkok, Thailand, December 2009
Proceedings, Part II

2 Part II

Springer

# Lecture Notes in Computer Science 5864

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

Chi Sing Leung   Minho Lee
Jonathan H. Chan (Eds.)

# Neural Information Processing

16th International Conference, ICONIP 2009
Bangkok, Thailand, December 1-5, 2009
Proceedings, Part II

Springer

Volume Editors

Chi Sing Leung
City University of Hong Kong
Department of Electronic Engineering
Hong Kong
E-mail: eeleungc@cityu.edu.hk

Minho Lee
Kyungpook National University
School of Electrical Engineering and Computer Science
1370 Sankyuk-Dong, Puk-Gu, Taegu, 702-701, Korea
E-mail: mholee@knu.ac.kr

Jonathan H. Chan
King Mongkut's University of Technology Thonburi
School of Information Technology
126 Pracha-U-Thit Rd., Bangmod, Thungkru, Bangkok 10140, Thailand
E-mail: jonathan@sit.kmutt.ac.th

Library of Congress Control Number: 2009939833

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

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

ISSN 0302-9743
ISBN-10 3-642-10682-X Springer Berlin Heidelberg New York
ISBN-13 978-3-642-10682-8 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12810987 06/3180 5 4 3 2 1 0

# Preface

This two-volume set constitutes the Proceedings of the 16th International Conference on Neural Information Processing (ICONIP 2009), held in Bangkok, Thailand, during December 1–5, 2009. ICONIP is a world-renowned international conference that is held annually in the Asia-Pacific region. This prestigious event is sponsored by the Asia Pacific Neural Network Assembly (APNNA), and it has provided an annual forum for international researchers to exchange the latest ideas and advances in neural networks and related discipline. The School of Information Technology (SIT) at King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand was the proud host of ICONIP 2009. The conference theme was "Challenges and Trends of Neural Information Processing," with an aim to discuss the past, present, and future challenges and trends in the field of neural information processing.

ICONIP 2009 accepted 145 regular session papers and 53 special session papers from a total of 466 submissions received on the Springer Online Conference Service (OCS) system. The authors of accepted papers alone covered 36 countries and regions worldwide and there are over 500 authors in these proceedings. The technical sessions were divided into 23 topical categories, including 9 special sessions. Technical highlights included a keynote speech by Shun-ichi Amari (the founder of APNNA); plenary and invited talks by Włodzisław Duch (President of the European Neural Network Society), Kunihiko Fukushima, Tom Gedeon, Yuzo Hirai (President of the Japanese Neural Network Society), Masumi Ishikawa, Nikola Kasabov (President of the International Neural Network Society), Minho Lee, Soo-Young Lee, Andrew Chi-Sing Leung, Bao-Liang Lu, Chidchanok Lursinsap, Paul Shaoning Pang, Ron Sun, Shiro Usui, DeLiang Wang, Jun Wang, Lipo Wang and Zhi-Hua Zhou. In addition, six tutorials by Włodzisław Duch, Chun Che Fung, Irwin King, Saed Sayad, Jun Tani and M. Emin Yuksel were part of ICONIP 2009. Also, for the first time, there was a Post-ICONIP Workshop held in a neighboring country to the host: the Workshop on Advances in Intelligent Computing (WAIC 2009) was held in Kuala Lumpur, Malaysia on December 7, 2009. Furthermore, the Third International Conference on Advances in Information Technology (IAIT2009) was collocated with ICONIP 2009.

We are indebted to the members of the conference Advisory Board as well as the Governing Board and Past Presidents of APNNA for their advice and assistance in the organization and promotion of ICONIP 2009. We are thankful to the Program Committee and Technical Committee members for their dedication and support in providing rigorous and timely reviews, especially for the last round of submissions due to our extended submission deadline. Each paper was reviewed by at least two referees and three or more reviews were provided in most of the cases. The Program Committee Chairs opted to use the relatively new OCS system and we put it through a rigorous workout and helped the system to smooth out numerous minor issues. We sincerely apologize for any inconvenience the authors may have experienced during the entire paper submission and reviewing process.

A special thanks to the Conference Secretariat, Olarn Rojanapornpun, who worked tirelessly to facilitate many of the conference delegates and to produce these final proceedings. The Organizing Committee members would like to express our sincere appreciation to the devoted behind-the-scene work by Wannida Soontreerutana, Chompoonut Watcharinkorn, Paweena Mongkolpongsiri, Thanyapat Natwaratit, Chutikarn Hongpitakkul, Korakot Eadjongdee, Suda Kasikitsakunphon, Kanittha Charoensuk and Monthana Hunjinda. Last but not least, the organizers gratefully acknowledge the contribution and support from all speakers, panelists and authors, as well as all other participants, in making ICONIP 2009 a resounding success.

December 2009

Jonathan H. Chan
Chi Sing Leung
Minho Lee

# Organization

## Organizer

School of Information Technology (SIT), King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand.

## Sponsor

Asia Pacific Neural Network Assembly (APNNA)

## Technical Co-sponsors

International Neural Network Society (INNS)
Japanese Neural Network Society (JNNS)
European Neural Network Society (ENNS)
IEEE Computational Intelligence Society (IEEE CIS)

## Conference Committee

| | |
|---|---|
| Honorary Chair | Shun-ichi Amari, Japan |

Advisory Board

Irwin King, Hong Kong
Nikola Kasabov, New Zealand
Soo-Young Lee, Korea
Derong Liu, USA
Nikhil R. Pal, India
Shiro Usui, Japan
Tom Gedeon, Australia
Takeshi Yamakawa, Japan
Włodzisław Duch, Poland
Lipo Wang, Singapore
Jun Wang, Hong Kong

| | |
|---|---|
| Local Steering Committee | Borworn Papasratorn, Thailand<br>Wichian Chutimaskul, Thailand<br>Chakarida Nulkoolkit, Thailand |
| General Chair | Jonathan H. Chan, Thailand |
| Program Chair and Co-chair | Andrew Chi-Sing Leung, Hong Kong<br>Minho Lee, Korea |
| Local Organizing Chair | Kittichai Lavangnananda, Thailand |

| | |
|---|---|
| Special Sessions Chairs | Masumi Ishikawa, Japan<br>Tom Gedeon, Australia<br>Shaoning Pang, New Zealand |
| Workshops Chairs | Lai Weng Kin, Malaysia<br>Chee-Peng Lim, Malaysia |
| Tutorials Chair | Kevin Wong, Australia |
| Competitions Chair | John Sum, Taiwan |
| Publicity Chair | Suree Funilkul, Thailand |
| Publication Chair | Kriengkrai Porkaew, Thailand |
| Posters and Demonstration Chair | Vithida Chongsuphajaisiddhi, Thailand |
| Local Arrangements Chair | Vajirasak Vanijja, Thailand |
| Conference Secretariat | Olarn Rojanapornpun, Thailand |

## Program Committee

Shigeo Abe, Japan
Siu Yeung Cho, Singapore
Doo-Hyun Choi, Korea
Tommy W.S. Chow, Hong Kong
Andrzej Cichocki, Japan
Ke-Lin Du, Canada
Kaizhu Huang, China
Daijin Kim, Korea
Seong Kong, USA
James Tin-Yau Kwok, Hong Kong
Soo-Young Lee, Korea
Frank Leung, Hong Kong
Bao-Liang Lu, China
Takashi Morie, Japan
Toshiaki Omori, Japan
Nikhil R. Pal, India
Hyeyoung Park, Korea
Naoyuki Sato, Japan
Dianhui Wang, Australia
Hau San Wong, Hong Kong
Haixuan Yang, China
Byoung-Ju Yun, Korea
Kun Zhang, Finland
Sabri Arik, Turkey
Yoonsuck Choe, USA
Seungjin Choi, Korea
Fu Lai Chung, Hong Kong
Kenji Doya, Japan
Tom Gedeon, Australia
Masumi Ishikawa, Japan
Sungshin Kim, Korea
Takio Kurita, Japan
Seong-Whan Lee, Korea
Yun-Jung Lee, Korea
Chunshien Li, Taiwan
Bob McKay, Korea
Takashi Omori, Japan
Seiichi Ozawa, Japan
Shaoning Pang, New Zealand
Jagath C. Rajapakse, Singapore
John Sum, Taiwan
Young-Gul Won, Korea
Zenglin Xu, Germany
Zhirong Yang, Finland
Zhigang Zeng, China
Liming Zhang, China

## Technical Committee

Shotaro Akaho
Tetsuya Asai
Hideki Asoh
Sang-Woo Ban
Hong Bao
Cesar Caiafa
B Chandra
Atthawut Chanthaphan
Shen Chen
Songcan Chen
Xi Chen
Seong-Pyo Cheon
Eng Yeow Cheu
Andrew Chiou
Heeyoul Choi
Ho-Hyoung Choi
Ji Ryang Chung
Zhaohong Deng
Yongtae Do
Bin Dong
Justin Dauwels
Hiroshi Dozono
Béla A. Frigyik
Minoru Fukumi
GH Gan
Eikou Gonda
Raj Gupta
Rodolfo Haber
Hisashi Handa
Osman Hassab Elgawi
Ken Hawick
Hanlin He
Xingshi He
Zhashui He
Akinori Hidaka
Hee-Dong Hong
Jin-Hyuk Hong
Antti Honkela
Xiaolin Hu
Xuelei Hu
Wei Huang
Yung-Fa Huang
Hiroaki Inayoshi
Hirotaka Inoue
Masato Inoue
Rolly Intan
Takaichi Ito
Tae-Seok Jin
Martin Johnson
Norhaslinda Kamaruddin
Keisuke Kameyama
Dae-Seong Kang
Hyohyeong Kang
Satoru Kato
Fujimura Kikuo
Jung-Gu Kim
Kye-Hyeon Kim
Kwang-Baek Kim
Sung hin Kim
Taesu Kim
Yong-Deok Kim
Yong-Tae Kim
YoungIk Kim
Yun-Ho Ko
Takanori Koga
Markus Koskela
Ryosuke Kubota
Jaerock Kwon
Johnny Lai
Choon Young Lee
Daewon Lee
Hyung-Soo Lee
Kwanyong Lee
Nung Kion Lee
Sang-Woong Lee
Gang Li
Gary C. L. Li
Xi Li
Xiangming Li
Xiao-Dong Li
Yufeng Li
Yen-Lun Liang
Chee Peng Lim
Dudy Lim
Heejin Lim
Kil-Taek Lim
Naiyan Lima
Iuon-Chang Lin
Wilfred Lin
Qingshan Liu
Rujie Liu
Weixiang Liu
Yu Liu
Zhiyong Liu
Danniel Cavalcante Lopes
Jacek Mańdziuk
Timothy Mann
Ohkita Masaaki
Seiji Miyoshi
Kenji Nagata
Mitsuteru Nakamura
Wakako Nakamura
Hidehiro Nakano
Yok Yen Nguwi
Qiang Ni
Kenji Nishida
Ikuko Nishikawa
Tohru Nitta
Richard Jayadi Oentaryo
Tetsuya Onoda
Matashige Oyabu
Tomoko Ozeki
Han-Saem Park
Hyung-Min Park
Kiyoung Park
Lae-Jeong Park
Sehoon Park
Seongbae Park
Sunho Park
Ekachai Phaisangittisagul
Anh Huy Phan
Kriengkrai Porkaew
Santitham Prom-on
Masukur Rahman
Piyush Rai
Myung-Cheol Roh
Hosaka Ryosuke
Ryo Saegusa
Stefan Schliebs
Gourab Sen Gupta
Daming Shi
Hiroyuki Shimai
Mats Sjöberg
Kingkarn Sookhanaphibarn
Shiliang Sun

Javan Tan
Dacheng Tao
Takashi Takahashi
Masahiro Takatsuka
Mieko Tanaka-Yamawaki
Teik Toe Teoh
Chan Wai Ting
Meng-Hsiun Tsai
Whye Loon Tung
Hiroaki Wagatsuma
Hiroshi Wakuya
Liang Wan
Bo-Hyeun Wang
Jun Wang
Rongjie Wang
Zhanshan Wang
Zhongsheng Wang
Kazuho Watanabe
Bunthit Watanapa
Ginny Wong
Arthur Liang-chuan Wu
Jing Wu
Jiunn-lin Wu
Wei Wu
Zhenping Xie
Hao Xiong
Lu Xu
Yang Xu
Tomoyuki Yamaguchi
Hee-Deok Yang
Huei-Fang Yang
Shengji (Sophie) Yao
Xucheng Yin
Noha Yousri
Yingwei Yu
Zhiwen Yu
Jingling Yuan
Shiu Yin Yuen
Jeong-Min Yun
Rafal Zdunek
Haijun Zhang
He Zhang
Shaohong Zhan
ZhanCheng Zhang
Lei Zheng

## Special Session Organizers

| | |
|---|---|
| Intelligent Data Mining | Kevin Wong |
| Data Mining for Cybersecurity | Tao Ban<br>Daisuke Inoue<br>Shaoning Pang<br>Youki Kadobayashi |
| Towards Brain-inspired Systems | Keiichi Horio |
| SOM and Related Subjects and its Applications | Nobuo Matsuda<br>Heizo Tokutaka<br>Masahiro Takatsuka |
| Neural Networks for Data Mining | Furao Shen<br>Zhi-Hua Zhou |
| Hybrid and Adaptive Systems for Computer Vision and Robot Control | Napoleon Reyes<br>Andre Barczak<br>Pitoya Hartono |
| Artificial Spiking Neural Systems: Nonlinear Dynamics and Engineering Applications | Toshimichi Saito<br>Hiroyuki Torikai |
| Computational Advances in Bioinformatics | Asawin Meechai<br>Santitham Prom-on |
| Evolutionary Neural Networks: Theory and Practice | Sung-Bae Cho<br>Kyung-Joong Kim |

## Local Sponsors

IEEE Thailand Section
Thailand Chapter of ACM
Software Park Thailand
Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology Association of Thailand (ECTI)
National Electronics and Computer Technology Center (NECTEC)

# Table of Contents – Part II

## Computer Vision

## Control and Robotics

## Evolutionary Computation

## Other Emerging Computational Methods

## Signal, Data and Text Processing

## Artificial Spiking Neural Systems: Nonlinear Dynamics and Engineering Applications

## Towards Brain-Inspired Systems

## Computational Advances in Bioinformatics

## Data Mining for Cybersecurity

## Evolutionary Neural Networks: Theory and Practice

## Hybrid and Adaptive Systems for Computer Vision and Robot Control

## Intelligent Data Mining

## Neural Networks for Data Mining

## SOM and Related Subjects and Its Applications

# Table of Contents – Part I

## Cognitive Science and Computational Neuroscience

## Neurodynamics

## Mathematical Modeling and Analysis

## Kernel and Related Methods

## Learning Algorithms

## Pattern Analysis

## Face Analysis and Processing

## Image Processing

## Financial Applications

# Obstacle Categorization Based on Hybridizing Global and Local Features

Jeong-Woo Woo[1], Young-Chul Lim[2], and Minho Lee[1,*]

[1] School of Electrical Engineering and Computer Science, Kyungpook National University,
1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701, Korea
[2] Division of Advanced Industrial Science & Technology, Daegu Gyeongbuk Institute of Science & Technology
711 Hosan-Dong, Dalseo-Gu, Taegu, 704-230, Korea
jw-woo@ee.knu.ac.kr, dgist@dgist.ac.kr, mholee@knu.ac.kr

**Abstract.** We propose a novel obstacle categorization model combining global feature with local feature to identify cars, pedestrians and unknown backgrounds. A new obstacle identification method, which is hybrid the global feature and local feature, is proposed for robustly recognizing an obstacle with and without occlusion. For the global analysis, we propose the modified GIST based on biologically motivated the C1 feature, which is robust to image translation. We also propose the local feature based categorization model for recognizing partially occluded obstacle. The local feature is composed of orientation information at a salient position based on the C1 feature. A classifier based on the Support Vector Machine (SVM) is designed to classify these two features as cars, pedestrians and unknown backgrounds. Finally, all classified results are combined. Mainly, the obstacle categorization model makes a decision based on the global feature analysis. Since the global feature cannot express partially occluded obstacle, the local feature based model verifies the result of the global feature based model when the result is an unknown background. Experimental results show that the proposed model successfully categorizes obstacles including partially occluded obstacles.

**Keywords:** Obstacle categorization, Modified GIST, Bottom-up saliency map model, C1-feature, Support Vector Machine.

## 1 Introduction

One of major causes of traffic accidents is a failure of obstacle detection due to driver's inattention and fatigue. Recently, the National Highway Traffic Safety Administration (NHTSA) reported that the 25-35% of the traffic accidents or 1.2 million vehicle crashes per year in the United Sated are resulted from distraction and inattention [1]. The researches on auto safety system have been highlighted through the $21^{st}$ century. Since, the vision-based technology can improve automotive safety and driver's convenience, especially, there are many published works concerning moving

[*] Corresponding author.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 1–10, 2009.

object detection for avoiding vehicle accidents [2-6]. These approaches use the information about symmetry [2], color [3], shadow [4], horizontal/vertical edge [5], and texture [6]. Among them, there is only focused to detect the presence of an obstacle or to detect the specific obstacle, such as a vehicle and a pedestrian. The obstacle categorization is an important task associated to the navigation of an intelligent vehicle. Upon detecting an obstacle, it is desirable that a categorization system is able to define which obstacle is in the pathway. The more obstacles categorization system is able to identify, the more intelligent vehicle comes true, for more different behaviors can be assigned to the intelligent vehicle. As an example, if an intelligent vehicle is only able to detect the presence of an obstacle without defining obstacle category, the only possible behavior is to deviate from an obstacle. In contrast, if the intelligent vehicle is able to identify that the obstacle is a pedestrian, as an example, it can be give the efficient avoidance path considering universal information of the pedestrians.

For the obstacle categorization, there are two methods that are commonly used for analyzing the image including an obstacle. The global feature based approaches are a simple way which can represent the obstacle image and they promise the credible performance than the local feature based approach when the obstacle is entirely included in an image without image translation and occlusion [7, 8]. However, the global feature based approaches are sensitive to the accuracy of the candidate localization. On the other hand, the local feature based approaches are known as having robustness to image translations and occlusions [9-11]. Thus, it is desirable that these two approaches are integrated to apply the obstacle categorization system in real-world problem such as cluttered backgrounds [12].

In this paper, we propose a vision-based obstacle categorization model combined the global and the local approaches for applying to a driver assistance system. For a global analysis of the obstacle image, a modified GIST algorithm with a SVM is used for categorizing an obstacle, which is robust to translation and size variation by using a biologically motivated C1 feature [11]. On the other hand, the local feature is composed of orientation information based on the C1 feature at the interesting areas selected by a biologically motivated bottom-up saliency map (SM) model [13]. Finally, the results of two approaches are combined. A result of the entire system is mainly followed by the result of the global feature based approach. If the result of the global feature based approach is an unknown background, however, the result of the entire system is followed by the result of the local feature based approach.

This paper is organized as follows. In Section 2, the description of proposed model is introduced. Experimental results are presented in Section 3. The conclusion and further works will be discussed in Section 4.

## 2 Proposed Models

The proposed obstacle categorization model is constructed based on global and local features. The global feature can represent an image by a compact way. Categorization performance of the global feature method is superior to the local feature based method when an object is well segmented [12]. Also, implementation of the global feature based method is easier than the local feature based identification because we need to consider more steps to design the local feature model such as the selection of local area and efficient representation scheme of the extracted features. However, the global feature is sensitive to occlusion and clutter and it require good segmentation. On the other hand, the local feature method does not need for segmentation and it is robust to occlusion and clutter. Thus, the proposed obstacle identification system is implemented by hybridizing a global feature based method as a role for primary decision with a local feature based algorithm as a supplementary role for additional verification to reduce the false-negative rate. Fig. 1 shows the overall procedure of the proposed obstacle identification system.

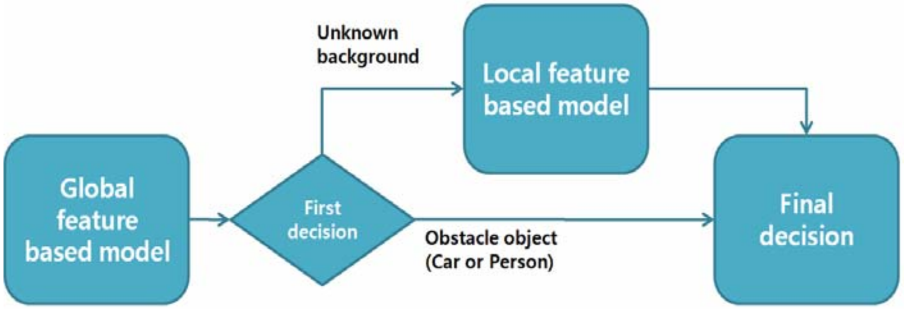

Unknown background
Local feature based model
Global feature based model
First decision
Obstacle object (Car or Person)
Final decision

**Fig. 1.** The procedure of an obstacle categorization system

### 2.1 Global Feature Based Model

The proposed global feature based model is based on a Modified–GIST algorithm that is robust to translation and size variation by using the biologically motivated C1 feature [11]. Fig. 2 shows the procedure of the global feature based model.

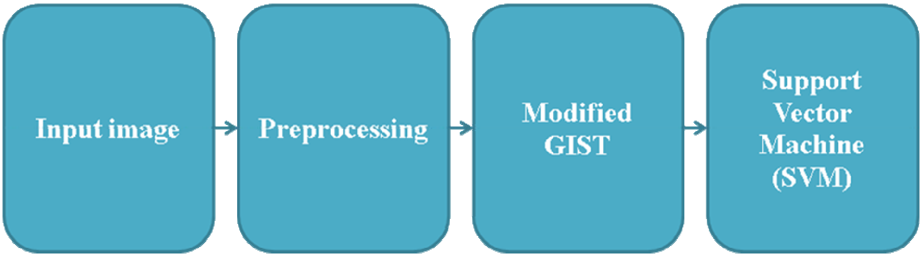


**Fig. 2.** The procedure of the global feature based model

The conventional GIST is the global feature method proposed by Torralba and A. Oliva (2006) [8]. The conventional GIST is too sensitive to occlusion and clutters in backgrounds. So, we improve the problem of conventional GIST by considering a C1 feature instead of using primitive orientation information. The C1 feature is orientation feature maps in the HMAX model. The HMAX model was proposed by Riesenhuber and Poggio (1999) [11], which is based on biologically motivated

The support vector machine (SVM) is used as a classifier, which is known as a powerful classifier to avoid overfitting by choosing a specific hyper plane among the many that can separate the data in the feature space [15].

## 2.2 Local Feature Based Model

The modified-GIST is still sensitive to partial occlusion and large translation even if the modified-GIST improves the robustness. Thus, we try to improve the obstacle identification performance by a local feature based model when an obstacle is seriously occluded or the segmented area by the stereo disparity map contains partial area of an object.

Fig. 4 shows the procedure of the local feature based model. We use a bottom-up SM model for selection of an interesting area to extract a local feature [13]. The bottom-up SM model is biologically motivated selective attention model. The roles of the retina cells and the LGN are reflected in the bottom-up SM model. The salient points are extracted by the SM using mask-off operation. And, the size of a salient region is calculated by the Entropy Maximization (EM) algorithm proposed by T. Kadir (2001) [16]. The conventional saliency map models consider several kinds of features such as intensity, edge, RG and BY opponents [13]. However, we consider only two feature maps such as intensity and edge, because the color information is not useful as the feature of an obstacle.

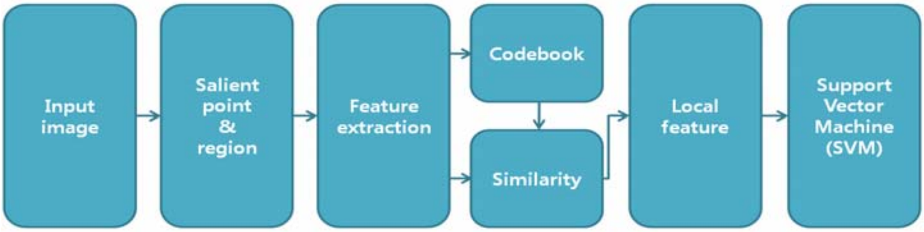


**Fig. 4.** The procedure of local feature based model

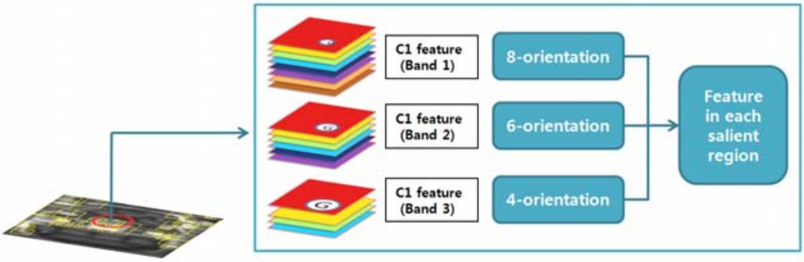


**Fig. 5.** The procedure of local feature based model

The feature of each salient region is extracted by same processing in the Modified-GIST. Because global and local feature based model use the same C1 feature, we can reduce computational complexity. Those features are used for generating the codebook. Fig. 5 shows the procedure of local feature extraction in each salient region

Since there are many local features in each image, we don't need to use those local features directly. So, we consider a codebook. In the learning process, local features generate a codebook. Similar codebook entries are eliminated above a certain threshold t. The similarity among codebook entries is measured by Pearson product-moment correlation coefficient (PMCC) [17] in Eq. (2).

$$r_{\mathbf{XY}} = \frac{\sum \mathbf{X}_i \mathbf{Y}_i - n\overline{\mathbf{X}}\overline{\mathbf{Y}}}{(n-1)s_{\mathbf{X}}s_{\mathbf{Y}}} = \frac{n\sum \mathbf{X}_i\mathbf{Y}_i - \sum \mathbf{X}_i \sum \mathbf{Y}_i}{\sqrt{n\sum \mathbf{X}_i^2 - (\sum \mathbf{X}_i)^2}\sqrt{n\sum \mathbf{Y}_i^2 - (\sum \mathbf{Y}_i)^2}} > t \tag{2}$$

where, **X** and **Y** are paired data sets. And $\overline{\mathbf{X}}$ is sample mean and s is sample standard deviation.

In our case, *t* is 0.8. The codebook entries are used as a basis to extract the local features of an image by comparing with local features in each salient region. The local feature of an image is composed of the maximum correlation coefficient of each codebook entry toward features of all salient regions. Eq. (3) is used to calculate the local feature of the image. The support vector machine (SVM) is also used as a classifier for the local feature based model.

$$\begin{gathered} \textit{Local Feature } f(C_i) = \max(r(X_j, C_i)),\ j = 1,2,3,\ldots,n \\ X\text{:}\textit{Features of all Salient region} \\ C_i\text{:}\textit{ i–th Codebook entries} \\ n\text{:}\textit{the number of Salient region} \\ r(X,Y)\text{:}\textit{correlation coefficient} \\ \max(X,Y) : \max \textit{ operation} \end{gathered} \tag{3}$$

## 3 Experimental Results

In order to evaluate the proposed obstacle identification system, we use the data on the CBCL database and Caltech database as shown in Fig. 6 [18, 19]. The train set consists of 900 images containing 300 images for codebook generation and 600 images for SVM learning. Three classes of objects such as cars, pedestrian and background are considered for training mode, in which each class has same number of data. The test set consists of 648 images with good segmentation of an object area and 400 images with partial occlusion. We use 216 images of each test class for evaluating the identification performance when an object is well segmented. In the test of occluded images, we use 200 images for each test class. We don't consider a partially occluded background image.

mechanism of visual cortex and composed of four hierarchical feed-forward layers, labeled as S1, C1, S2 and C2 named by analogy with the V1 simple and complex cells based on Hubel and Wiesel's paper (1968) [14]. The C1 feature is known as having robust characteristic about scale and translation.

For considering semantic information of the localized input image, the input image is divided into 4x4 sub-regions in the Modified-GIST. And three-layers of the C1 feature are generated by Gabor filters with various orientations and scales in each sub-region. In order to generate the C1 feature, we process two steps. The first step is extraction of S1 features, which corresponds to simple cells in V1. The S1 features are obtained by convolution of the input image and Gabor filters which are made by various orientation and scales. We use following Eq. (1) to acquire these Gabor filters.

$$G(x,y) = \exp\left(-\frac{(X^2+\gamma^2 Y^2)}{2\sigma^2}\right)\cos\left(\frac{2\pi}{\lambda}X\right) \qquad (1)$$

$$\textit{where}\ \ X = x\cos\theta - y\sin\theta,\ \ Y = x\sin\theta + y\cos\theta$$

For our case, we use Gabor filters with six different scales (7x7, 9x9, 11x11, 13x13, 15x15, 17x17). Layer 1, consisting of 8-orientation information, is calculated when scales of Gabor filter are 7x7 and 9x9, layer 2, consisting of 6-orientation information is calculated when scales of Gabor filter are 11x11 and 13x13, respectively, and layer 3, consisting of 4-orientation information is calculated when scales of Gabor filter are 15x15 and 17x17. The second step is to generate C1 features from S1 features. For each layer, sub-sampling is applied by taking the maximum value over a 50% overlapped grid with cells of different sizes (layer 1: 8x8, layer 2: 10x10, layer 3: 12x12). Therefore, it is possible to obtain three-layers of the C1 feature.

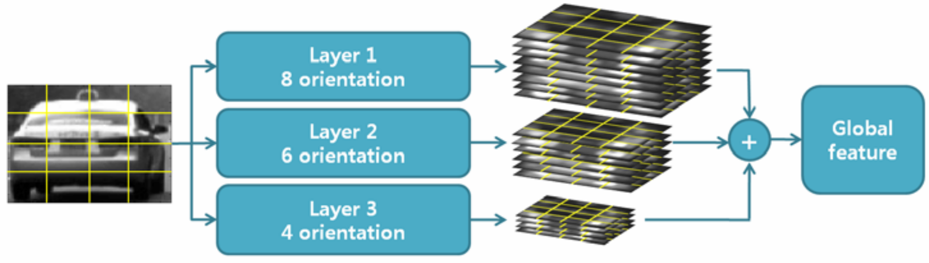
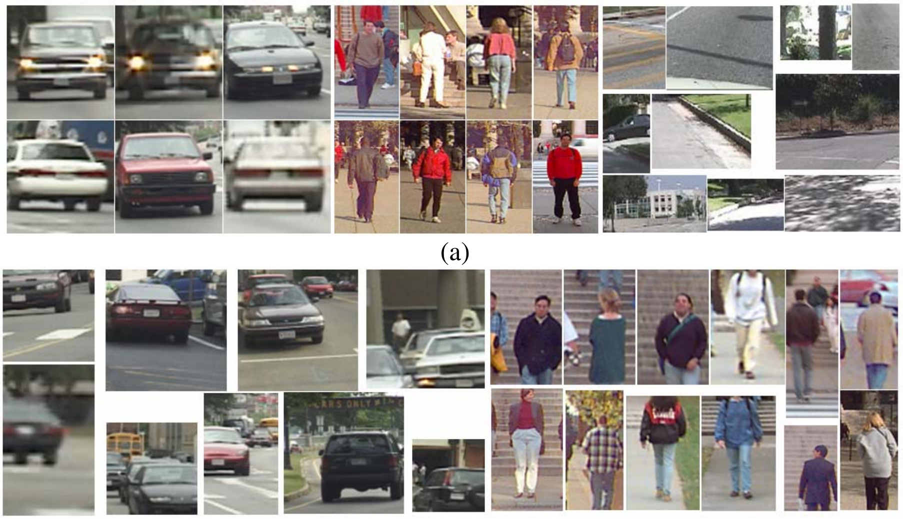


**Fig. 3.** The procedure of global feature extraction using the modified GIST

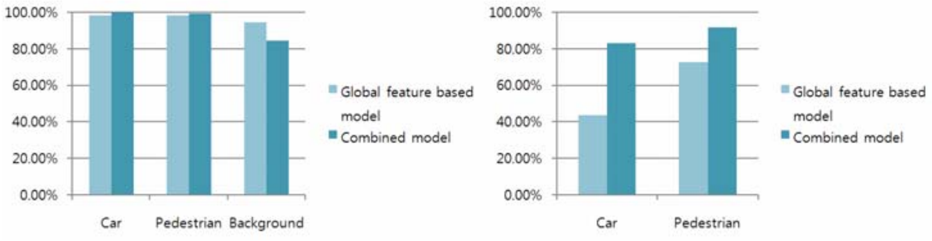
The global feature is composed of the average C1 value of each sub-region. The modified GIST is more robust than the conventional GIST when an object is translated within a few pixels, because C1 feature corresponding to complex cells that have some tolerance to translation and size variance, which is obtained by using a max pooling operation [11]. Fig. 3 shows the summaries for the procedure of global feature extraction using the modified-GIST.



(a)

(b)

**Fig. 6.** CBCL and Caltech databases which are used in the experiments. (a) Image set which is well segmented. (b) Image set which is partially occluded and translated.



(a) (b)

**Fig. 7.** The experimental results of the proposed obstacle identification system. (a) The experimental result when an object is well segmented. (b) The experimental result when an object is partially occluded.

Fig. 7 shows the experimental results of the proposed obstacle identification system. Categorization rates on the cars and the pedestrians increase from 98.15% to 100% by hybridizing method using global and local features. But, a categorization rate on background images decreases from 94.14% to 84.26%. In the driving assistance system, the false-negative result is more dangerous than the false-positive result because the false-negative result can lead to a traffic accident by driver's inattention. In the occluded images, the proposed method dramatically increases the identification performance as shown in Fig. 7 (b).

We also evaluate the effectiveness of the modified-GIST comparing with the conventional GIST. As shown in Fig. 8, categorization performance of the modified

GIST is superior to the conventional GIST. Although the categorization rates of the modified GIST are similar to ones of the conventional GIST for cars and pedestrians, the proposed method outperforms the convention GIST when the obstacles are miss-detected as backgrounds. This result means that the C1 feature enhance the robustness of the conventional GIST model. Also, it shows that the biologically motivated C1 feature is useful algorithm.

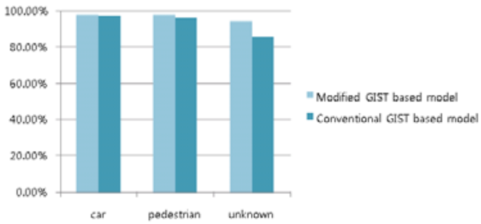


**Fig. 8.** Performance comparison result of the modified-GIST and conventional GIST model

In another experiments, we compare the proposed local feature based model with the HMAX model [11]. The proposed local feature based model shows comparable performance with the HMAX model for well segmented images as shown in Fig. 9 (a). But, the partially occluded obstacles are better identified by the proposed local feature method than the HMAX model as shown in Fig. 9 (b). The HMAX model uses 2000 prototype as bases, while the proposed local feature based model only uses 328 codebooks. Furthermore, codebook entries just consist of 18 dimensions vector.

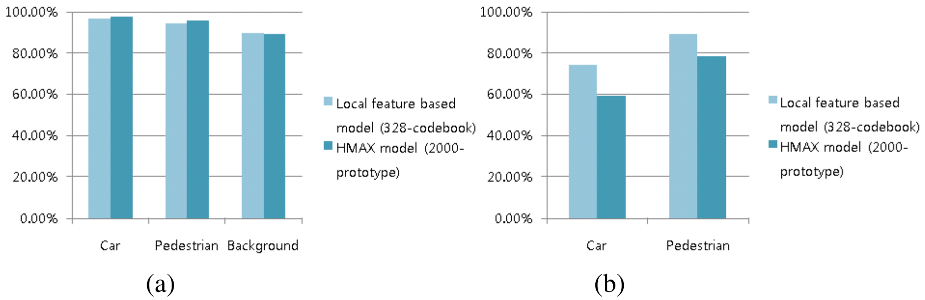


**Fig. 9.** Performance comparison results of the proposed modified-GIST based model and the local feature based model. (a) The comparison result between the modified-GIST and the conventional GIST when the obstacle is well segmented. (b) The comparison result between the proposed local feature based model and the HMAX model in the occluded condition.

## 4 Conclusion and Further Works

In this paper, we propose an obstacle categorization model based on hybridizing global and local features. The modified GIST is used for extracting a global feature.

And the local feature is generated by comparison the codebook with orientation information at a salient position based on the biologically motivated C1 features. The SVM is designed for classifying those global and local features, respectively. Our experiments illustrate that the global and local hybrid features for obstacle categorization model can improve categorization performance when the obstacle is partially occluded. Also, our experimental results show that the C1 feature is very useful to enhance the robustness in the object recognition problem.

In future, we need to verify the proposed algorithm in various circumstances including different weather conditions and night view scenes. Also, we plan to design the driver assistance system with proposed obstacle categorization model, by incorporating stronger obstacle detector in real driving scenes.

**Acknowledgments.** This research was supported by the Daegu Gyeongbuk Institute of Science and Technology (DGIST), the development of real-time environment recognition techniques for moving objects.

## References

1. National Highway Traffic Safety Administration, USA, `http://www.nhtsa.dot.gov`
2. Kuehnle, A.: Symmetry-based recognition for vehicle rears. Pattern Recognit. Lett. 90(7), 1258–1271 (2002)
3. Buluswar, S.D., Draper, B.A.: Color machine vision for autonomous vehicles. Int. J. Eng. Appl. Artif. Intell. 1(2), 245–256 (1998)
4. Dickmanns, E., et al.: The seeing passenger car 'vamors-p'. In: Proc. Int. Symp. Intelligent Vehicles, pp. 24–26 (1994)
5. Goerick, C., Detlev, N., Werner, M.: Artificial neural networks in real-time car detection and tracking applications. Pattern Recognit. Lett. 17, 335–343 (1996)
6. Kalinke, T., Tzomakas, C., Seelen, W.V.: A texture-based object detection and an adaptive model-based classification. In: Proc. IEEE Int. Conf. Intelligent Vehicles, pp. 143–148 (1998)
7. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cogn. Neurosci. 3, 71–86 (1991)
8. Oliva, A.: Gist of the Scene. J. Neurobiol. Atten. CA, 251–256 (2005)
9. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. Compt. Vis. Pattern Recog., pp. 511–518 (2001)
10. Leibe, B., Leonardis, A., Schiele, B.: Combined Object Categorization and Segmentation with an Implicit Shape Model. In: Proc. Euro. Conf. Compt. Vis., Prague (2004)
11. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. J. Neurosci. 2, 1019–1025 (1999)
12. Lisin, D.A., Matter, M.A., Blaschko, M.B.: Combining Local and Global Image Features for Object Class Recognition. In: CVPR (2005)
13. Jeong, S., Ban, S.-W., Lee, M.: Stereo Saliency Map Considering Affective Factors and Selective Motion Analysis in a Dynamic Environment. J. Neural Netw. 21(10), 1420–1430 (2008)
14. Huber, D.H., Wiesel, T.N.: Receptive fields and functional architecture of monkey striate cortex. J. Physiol. 148, 574–591 (1968)
15. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. J. Data Min. Knowl. Discov. 2, 121–167 (1998)

16. Kadir, T., Brady, M.: Saliency, Scale and Image Description. Int. J. Computer Vis. 45(2), 83–105 (2001)
17. Rodgers, J.A., Nicewander, W.A.: Thirteen ways to look at the correlation coefficient. J. Amer. Statistician. 42(1), 59–66 (1988)
18. Center for Biological & Computational Learning. MIT, Cambridge, http://cbcl.mit.edu
19. Computational Vision Lab, Caltech, http://www.vision.caltech.edu

# Defect Detection and Classification in Citrus Using Computer Vision

Jose J. Lopez, Emanuel Aguilera, and Maximo Cobos

Institute of Telecommunications and Multimedia Applications (iTEAM),
Universidad Politécnica de Valencia
Camino de Vera s/n. 46022, Valencia, Spain
{jjlopez,emagmar,mcobos}@iteam.upv.es

**Abstract.** In this paper, a system for quality control in citrus is presented. In current citrus manufacturing industries, calliper and color are successfully used for the automatic classification of fruits using vision systems. However, fault detection in the citrus surface is carried out by means of human inspection. In this work, a computer vision system capable of detecting defects in the citrus peel and also classifying the type of flaw is presented. First, a review of citrus illnesses has been carried out in order to build a database of digitalized oranges classified by the kind of fault, which is used as a training set. The segmentation of faulty zones is performed by applying the Sobel gradient to the image. Afterwards, color and texture features of the flaw are extracted, some of them related with high order statistics. Several techniques have been employed for classification purposes: Euler distance to a prototype, to the nearest neighbor and k-nearest neighbors. Additionally, a three layer neural network has been tested and compared, obtaining promising results.

**Keywords:** Computer vision, Automatic inspection system, Texture analysis segmentation, Quality control.

## 1 Introduction

The citrus industry in Spain is very important, being one of the biggest ones in the world regarding production, with about 6 million tons in the year 2008. This quantity includes all the range of citrus: orange, lemon, grapefruit and a great number of tangerine varieties. Today, machine vision has been introduced in many industrial applications for citrus processing, allowing the automation of tasks performed so far by human operators. These systems can work faster than manual procedures, but not better in all the cases. Although most of the sorting and packing stages in packing houses are automated, workers are still needed to discard the fruits that present defects in their peel. The quality of the peel in fruits, which should be free from defects, is one of the most influential factors in the price of fresh fruit. The reason lies in the association that consumers make between good appearance and quality.

Considerable effort has been made in the field of machine-vision-based defect sorting and grading for different fruits as citrus [1][2] , apples [3][4], olives [5],

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 11–18, 2009.

potatoes [6], other fruits and industries [7]. Nowadays, the increase in computer power at affordable prices and the introduction of multiple core processors allows to process complex images in a short time and to use more complex algorithms. A very important part of the process is calyx localization, mainly for two reasons. On one hand, it is needed to distinguish the calyx (that it is not a defect) from other peel defects [8]. On the other hand, there is an increasing demand on automatic systems for rotating the fruit to the correct calyx-relative position. This makes possible to use other fruit processing systems, like industrial peelers.

As commented above, there exist other references related to the detection of defects in the citrus peel [1]. However, in these works, the classifier is only capable to inform whether there is a defect on the peel or not, thus disregarding the information about the defect type. As an extra advantage, it is very interesting to distinguish between different defect types, since this can help to decide on the most suitable actions that should be carried out with the faulty piece of fruit. In particular, some defects do not forbid the fruit to be taken to the market, but they impede it to belong to a higher category. On the other hand, other defects make the fruit be directed to juice manufacturing or, in case of a serious defect, the fruit is directly thrown away.

### 1.1 Objectives

The aim of this work is to develop several image analysis algorithms in order to classify different surface defects in citrus, including the calyx detection. This objective is very ambitious, since the defect type in fruits peel can be related to a high number of different diseases, such as fungus, insects bites or other animals attacks. Moreover, defect types are also dependent on the geographical region where the citrus is grown. In order to establish some limitations, we have focused on a finite and affordable number of existing types. To achieve this objective, a system made up of three steps is proposed:

- First, a segmentation stage based on boundary detection using a Sobel gradient mask for defect localization is explained.
- The second stage is the feature extraction of the defect area found. Five properties on RGB color space are proposed to distinguish between different peel faults.
- In the last stage, two classification techniques are applied over the features: distance-based classifiers and neural networks.

Previously to these three stages, it will be necessary to create an image data base containing all the defect types manually classified by an expert.

## 2 System Set-Up Overview and Citrus Defect Database

Although the final aim of the system here presented is to be included in a conveyor belt, for all the experiments conducted in this work, the fruits were kept static inside an inspector chamber. This had diffuse lighting controlled by our lab. In order to obtain images from all the orange surface, it was manually rotated for obtaining 4 different captures, covering the 100% of the surface. However, when moving from laboratory conditions to a real industry situation, it will be necessary to consider that the fruits are carried by a conveyor-belt from the unload area to the classification area. The camera will be placed at some point of the conveyor.

A common shutter speed of 1000 can be employed without blur degradation. However, since high shutter speeds are employed, high illumination is needed. To avoid the problems caused in the segmentation step by highlights on the fruit surface, we used diffused light. White bulbs where used better than transparent ones because in the first the filament is hidden. Approximately 600 W were needed to provide the correct illumination. Fluorescent light was discarded because of the noise it produced. This was checked in a real industry conveyor belt observing that, since this artificial light is not really white, the camera had to be previously equalized with white balance control. The cameras employed where digital cameras with direct digital output via *Firewire*. The resolution employed was 640 × 480 pixels at 256 levels per channel (R,G,B).

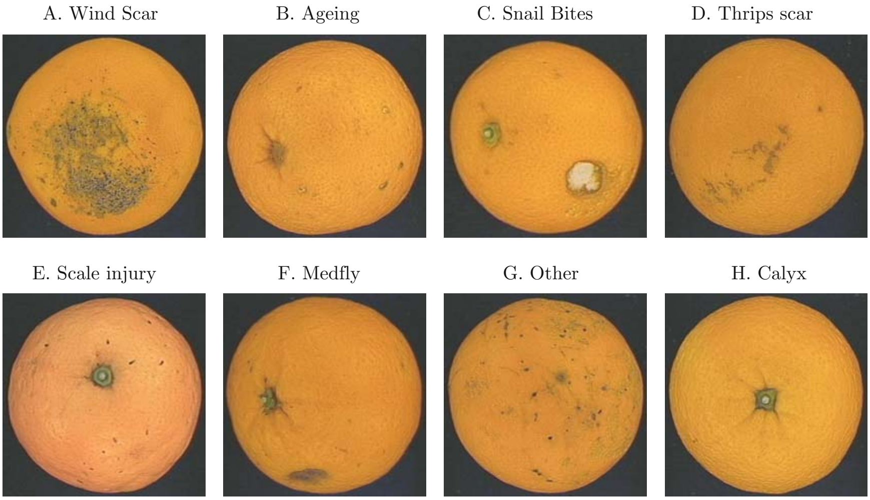
### 2.1 Citrus Defect Database

A database of the most common different defects in the eastern region of Spain was built. A set of 150 oranges from "*navel-late*" and "*valencia-late*" varieties were digitalized in the laboratory. Each orange was captured in 10 different positions covering the full surface, obtaining a set of 1500 images.

Each orange was classified by an expert and afterwards labelled according to the defect type its peel had. This data base constituted the training set for the classifying stage. The defects that were identified and that were considered for present work are the following:

- A. "Wind scar" : the wind can produce lights lesions due to friction with branches.
- B. "Stem-end breakdown" or "ageing".
- C. "Snail bites".
- D. "Thrips scar".
- E. "Scale injury". Where we can find "Parlatoria ziziphi", "Parlatori pergandei", "Chrysomphalus dictiospermi", etc.
- F. "Mediterranean fruit fly".
- G. Others faults that can be located from visual inspection.
- H. Stem-calyx. The location of the calyx is an important step, because it must not be confused with one defect. Also, the zone where a calyx was removed must not be confused either. These considerations can be found in some other works [9][8].

Figure 1 shows different examples of the analyzed defect groups (including the Stem-calyx).



A. Wind Scar
B. Ageing
C. Snail Bites
D. Thrips scar
E. Scale injury
F. Medfly
G. Other
H. Calyx

**Fig. 1.** Example of different types of peel defects

# 3 Proposed Processing

## 3.1 Defect Area Segmentation

Assuming a digitalized RGB color image of a citric, the first objective is the segmentation of the peel defect area. We propose a boundary detection method for this task based on a Sobel gradient mask.

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{1}$$

$$Gradient = \sqrt{(|G_x| + |G_y|)}. \tag{2}$$

For the most part of fruits, as oranges, the components R and G of the surface color are always stronger than the B component, as it can be seen in Figure 2. However, the G component can sometimes provide wrong cues about the defect of the fruit, because the mature state of different parts of the fruit can be interpreted as boundary. So it seems reasonable to use the R component in this phase.

After boundary detection, the resulting image must be post-processed in order to recognise the different boundaries (orange contour and faults outline). Each edge is assigned to a label. To assign a new edge, neighborhood and gradient threshold criterions are used. The process of segmenting the orange from the background is not difficult. We only consider the main edge, thus the background can be non uniform as the conveyor belt. Since each defect zone is limited by a labelled edge, we can locate it inside a polygon approximation, see Figure 3(a), so

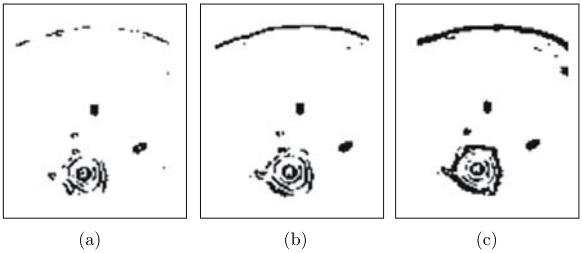


**Fig. 2.** Images after Sobel gradient mask and thresholding. a)Blue , b)Green and c) Red.

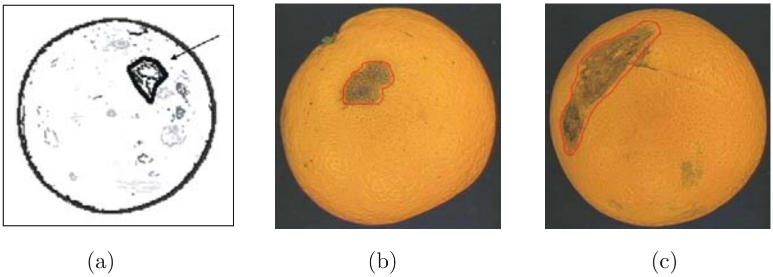


**Fig. 3.** (a) Polygonal approximation of one faulty zone. Each grey level stands for a different border. (b-c) Segmentation area for two types of defects.

that we avoid problems of edge discontinuity. Also we can determine an approximation of the percent of damaged zone over the total orange surface. Finally, Figure 3(b-c) shows the result of the segmentation area marked on the original image by a red line. It can be seen that the segmentation is very precise. This fact is very important for the application of the following defect classification stage.

### 3.2 Feature Extraction

In this second stage, we propose five features based on first, second and high order statistics, that let us distinguish between different types of peel defect regions:

- Mean:

$$\mu[C] = \sum_{l=0}^{255} l \cdot P(l, C), \tag{3}$$

  where $C$ indicates the R,G or B component and $P(l, C)$ the histogram of each color component [10].
- Variance

$$\sigma^2[C] = \sum_{l=0}^{255} (l - \mu)^2 \cdot P(l, C), \tag{4}$$

– Skewness

$$s[C] = \sum_{l=0}^{255} (l - \mu)^3 \cdot P(l, C)/\sigma^{3/2} \tag{5}$$

– Kurtosis

$$k[C] = \sum_{l=0}^{255} (l - \mu)^4 \cdot P(l, C)/\sigma^2 \tag{6}$$

– Range

$$R[C] = \frac{\max_{0<l<255}\{P(l, C) <> 0\} - \min_{0<l<255}\{P(l, C) <> 0\}}{255} \tag{7}$$

These features are computed for each R, G and B component so we will have a vector of 15 features for the next classification stage.

### 3.3 Classification

After using the previously presented segmentation techniques, some different faulty areas are obtained. For these areas, the above considered features are extracted in order to conform the feature vector. A major task after the feature extraction is to classify the segmented area into one of several fault categories, including the calyx detection. For this purpose, we are employing mainly two classification techniques: distance-based classifiers and neural networks.

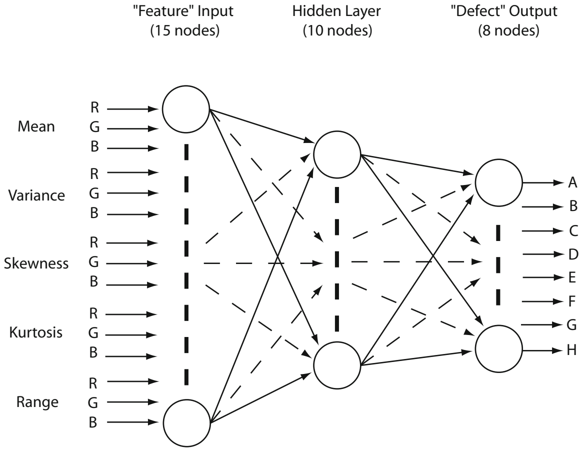


**Fig. 4.** Three layer neural network employed in the classifier

With distance-based classifiers we are employing three different methods: the mininum distance (MD), the nearest neighbour (NN) and the k-nearest neighbours (k-NN). While the first one selects the unique prototype minimum distance, the second one selects the nearest training sample. The third one selects the k-nearest training samples. Also, different feature combinations are used. The first test uses mean and range, the second test uses variance, skewness and kurtosis, and a mixture of all these features is used in the third test.

A feedforward backpropagation neural network (Net) is the other technique being used to classify the different fault categories. The network model designed has 15 input nodes, one for each of the features extracted in the previous section. Then, it has one hidden layer with ten nodes. And finally it has eight output nodes, one for each kind of citrus fault, included the calyx detection, see Figure 4. All the functions to activate each neurone are considered to be sigmoids. The neural network was trained by back-propagation until reaching a correct classification.

## 4 Test and Results

Using the system discussed before, we digitalized another different set of 180 oranges from the "*navel-late*" and "*valencia-late*" varieties. In this case each orange was digitalized only in 4 different positions simulating a conveyor belt, obtaining a set of 720 images. Also in this case, the expert previously classified the type of defect in order to evaluate the performance achieved by the developed automatic system. Next, the proposed computer vision software was executed and feeded with the second database in order to classify this new set of oranges. Several classification tasks were performed using the four proposed methods and the training data set. It was also studied how the performance was affected depending on the chosen features. With this goal, three different feature sets were created, using one or more feature vectors (FV) with each of the proposed classifiers.

– FV1: (Mean, Range)
– FV2: (Variance, Skewness, Kurtosis)
– FV3: (Mean, Range, Variance, Skewness, Kurtosis)

Columns reflect the classification method: and the feature vector used. Rows reflect the kind of citrus defect as described in section 2.1. Each cell shows the correct detection rate. Analyzing the table values, it can be observed that the use of more features results in an increased performance, which justifies the introduction of high order features present in FV2. Regarding the classification algorithm, it is also noticeable that the more sophisticated ones (k-Nearest Neighbors and Neural Network) produced considerably better results.

## 5 Conclusions and Future Work

A system for quality inspection in the citrus industry has been presented. In its design, a special effort has been made for building a reliable training data set including many types of peel flaws. This defect identification task has been carried out by experts in the field, providing our system with very useful information successfully employed in the classification task. The system is able to run in real time

**Table 1.** Overall percentage correct classification

| | Prot. | | | NN | | k-NN | | | Net |
|---|---|---|---|---|---|---|---|---|---|
| | FV1 | FV2 | FV3 | FV1 | FV2 | FV1 | FV2 | FV3 | FV3 |
| A | <50 | 50 | <50 | <50 | 62.5 | <50 | 97.3 | <50 | 94.2 |
| B | <50 | 71.4 | 60.2 | 100 | 100 | 50 | 100 | 100 | 100 |
| C | 71.4 | <50 | 57.1 | 78’5 | 57.1 | 78.5 | 50 | 92.8 | 94.2 |
| D | 77 | <50 | 69.2 | 61’5 | 69.2 | 62.6 | 84.6 | 92.3 | 72.3 |
| E | 94.1 | 66.6 | 100 | <50 | 66.6 | <50 | <50 | <50 | 63.2 |
| F | <50 | <50 | <50 | 53’8 | 69.2 | 77.1 | 52.3 | 84.6 | 72.5 |
| G | 66.6 | <50 | <50 | <50 | 50 | <50 | <50 | 75 | 69.3 |
| H | <50 | <50 | <50 | <50 | 58.3 | <50 | 58’3 | 66.8 | 62.1 |

for a typical conveyor belt speed of 0.1 s. In our implementation, a quad-core state-of-the-art personal computer was used. Our future work is focused on improving the performance by introducing new statistical-based features not only related to color. For example, the shape and size of defects is being currently exploited as a possible improvement line.

## References

1. Blasco, J., Aleixos, N., Molto, E.: Computer vision detection of peel defects in citrus by means of a region oriented segmentation algorithm. Journal of Food Engineering 81, 535–543 (2007)
2. Blasco, J., Aleixos, N., Gomez, J., Molto, E.: Citrus sorting by identification of the most common defects using multispectral computer vision algorithm. Journal of Food Engineering 83, 384–391 (2007)
3. Wen, Z., Tao, Y.: Building a rule-based machine-vision system for defect inspection on apple sorting and packing lines, vol. 16, pp. 307–313 (1999)
4. Qingzhong, L., et al.: Computer vision based system for apple surface defect detection. Computer and Electronics in Agriculture 36, 215–223 (2002)
5. Diaz, R., Gil, L., Serrano, C., Blasco, M., Molto, E., Blasco, J.: Comparison of three algorithms in the classification of table olives by means of computer vision. Journal of Food Engineering 61, 101–107 (2004)
6. Muir, A.Y., Porteus, R.L., Wastie, R.L.: Experiments in the detection of incipient diseases in potato tubers by optical methods. Journal of Agricultural Engineering Research 27, 131–138 (1982)
7. Conners, R.W., et al.: Identifying and Locating Surface Defects in Wood. IEEE Transactions on Pattern Analysis and Machine Intelligence 5, 573–583 (1983)
8. Ruiz, L.A., Molto, E., Juste, F., Pla, F., Valiente, R.: Location and Characterization of the Stem-Calyx Area on Oranges by Computer Vision, vol. 64, pp. 165–172 (1996)
9. Molto, E., et al.: A computer vision system for inspecting citrus,peaches and apples. In: Proceedings of VII National Symposium on Pattern Recognition and Image Analysis, pp. 121–126. Sabadell, Barcelona (1997)
10. Tan, T.S.C., Kittler, J.: Colour Texture Analysis Using Colour Histogram. In: IEEE Proc.-Vis. Image Signal Process, vol. 141 (1994)

# Superresolution from Occluded Scenes

Wataru Fukuda, Atsunori Kanemura, Shin-ich Maeda, and Shin Ishii

Graduate School of Informatics, Kyoto University, Kyoto 611-0011, Japan
{fukuda,atsu-kan,ichi,ishii}@sys.i.kyoto-u.ac.jp

**Abstract.** We propose a Bayesian image superresolution method that estimates a high-resolution background image from a sequence of occluded observations. We assume that the occlusions have spatial and temporal continuities. Such assumptions would be plausible, for example, when satellite images are occluded by clouds or when a tourist site is obstructed by people. Although the exact inference of our model is difficult, an efficient superresolution algorithm is derived by using a variational Bayes technique. Experiments show that our superresolution method performs better than existing methods that do not assume the occlusions or that assume the occlusions but do not assume the temporal continuities of the occlusions.

**Keywords:** Occlusion removal, image superresolution, variational Bayes.

## 1 Introduction

Superresolution is an image processing method that estimates a high-resolution (HR) image from a sequence of low-resolution (LR) images of a single scene [1,2,3,4]. In this study, we focus on a superresolution method that deals with occluded observations. Occlusions are typical, for example, in satellite images, where clouds can obstruct the land surface, or in pictures taken at tourist sites, where people may disturb taking a clean image.

There are two strategies to superresolving an HR image from given occluded LR images. One way is to identify the occlusions and use the unoccluded regions of the observations for superresolution [1,5] and the other is to employ a robust cost function that automatically discards inconsistent parts of the observations [2]. The former, which we adopt in this study, has several advantages over the latter; it can incorporate explicit prior knowledge about the occlusion and clean HR image, and desirable properties on them can be incorporated. Furthermore, by representing the existence of occlusions in a probabilistic manner, it can produce better estimates of the clean HR image by incorporating the uncertainty whether a pixel in an observed image is occluded or not.

The estimation of a clean HR image and the detection of occlusions are closely related; if we know one, we are able to accurately estimate the other. Consequently, a separated treatment of the estimation of the clean HR image and the identification of the obstructers (e.g. as in [6]) would prevent us from accurate estimation. Therefore, we conduct their simultaneous estimation based

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 19–27, 2009.

on a consistent probabilistic model incorporating prior knowledge about the HR image and occlusions; in other words, we estimate posterior distributions of the clean HR image and obstructers conditioned on given observations based on a joint distribution of the clean HR image, the obstructers, and the given observations. However, exact Bayesian inference in our model is difficult due to the high dimensionality of the variables; thus we employ variational Bayes and Laplace approximation to obtain a practical algorithm.

## 2 Problem Formulation

Suppose we have $T$ LR images $\mathbf{y} = \{\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(T)}\}$ that are consecutive and partly occluded observations of the same scene, where $\mathbf{y}^{(t)}$ is a vector of length $P_{\mathrm{L}}$ obtained by raster-scanning of the image. Our aim is to reconstruct an HR image $\mathbf{x}$, which is a vector of length $P_{\mathrm{H}}$ $(> P_{\mathrm{L}})$ from its observations $\mathbf{y}$.

The observed images are assumed to be generated from the HR image by i) applying shift and rotation motions, ii) convolving with a point spread function (PSF), iii) downscaling, and iv) adding Gaussian noise. To represent possible occlusions, we further assume the variance of the noise process is space-*variant* depending on the existence of occluders [1]. Given the occluders and $\mathbf{x}$, $\mathbf{y}^{(t)}$ is assumed to be generated by the following linear equation:

$$\mathbf{y}^{(t)} = W^{(t)}\mathbf{x} + \boldsymbol{\epsilon}^{(t)} \quad \text{for } t = 1, \dots, T, \tag{1}$$

where $W^{(t)}$ denotes a $P_{\mathrm{L}} \times P_{\mathrm{H}}$ observation transformation matrix that reflects the assumptions i)–iii), and the noise term $\boldsymbol{\epsilon}^{(t)}$ obeys *non*-iid Gaussian distribution whose variance is space-variant according to the occlusion pattern. In this study, we assume for simplicity the matrices $W^{(t)}$ are known.

## 3 Hierarchical Bayesian Model

According to a Bayesian treatment of superresolution, the HR image is estimated by the mean of its posterior distribution $p(\mathbf{x}|\mathbf{y})$,

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{\int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})\,\mathrm{d}\mathbf{x}}. \tag{2}$$

We define a hierarchical model by introducing hidden variables $\mathbf{z} = \{\mathbf{z}^{(1)}, \cdots, \mathbf{z}^{(T)}\}$ that represent occlusions on the LR images. Each $\mathbf{z}^{(t)} \in \{-1, +1\}^{P_{\mathrm{L}}}$ has the same length as the corresponding LR image $\mathbf{y}^{(t)}$, and $z_i^{(t)} = -1$ denotes that the $i$th pixel of $\mathbf{y}^{(t)}$ is occluded, whereas $z_i^{(t)} = +1$ denotes the corresponding pixel is intact. Therefore, the two-dimensional configuration of $\mathbf{z}^{(t)}$ shows the shape of the occlusion on the $t$th LR image. We further introduce hidden variables $\boldsymbol{\theta}$ which represent the temporal movements of the occluder. In the following subsections we define probability distributions that constitute our model (2).

### 3.1 Observation Model

The observation process is probabilistically given by

$$p(\mathbf{y}|\mathbf{x},\mathbf{z}) = \prod_{t=1}^{T} \mathcal{N}(\mathbf{y}^{(t)}|W^{(t)}\mathbf{x}, B(\mathbf{z}^{(t)})^{-1}), \tag{3}$$

where $B(\mathbf{z}^{(t)})$ is a $P_{\mathrm{L}} \times P_{\mathrm{L}}$ diagonal precision matrix whose elements $\beta_i(z_i^{(t)})$ $(i = 1, \ldots, P_{\mathrm{L}})$ are defined by

$$\beta_i(z) = \begin{cases} \beta_{\mathrm{H}}, & z = +1, \\ \beta_{\mathrm{L}}, & z = -1. \end{cases} \tag{4}$$

We assume $\beta_{\mathrm{H}} > \beta_{\mathrm{L}}$ so that $z_i^{(t)} = +1$ indicates high reliability (unoccluded) on the $i$th pixel of the $t$th observation whereas $z_i^{(t)} = -1$ low reliability (occluded). This space-variant variance model is the same as [1].

### 3.2 Image Prior

We assume a Gaussian image prior for the HR image

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, (\rho A)^{-1}), \tag{5}$$

where the matrix $A$ is a first-order difference operator and $\rho$ is a scalar that controls the strength of smoothness. This prior represents our *a priori* knowledge that the HR image would be spatially smooth.

### 3.3 Prior for Occlusion Patterns

In order for a prior to represent the uniform motion of the occlusions, we assume a Markov property of the motion between the consecutive observations and introduce a two-dimensional vector $\boldsymbol{\theta}^{(t)} = [\theta_1^{(t)}, \theta_2^{(t)}]^\top$ that denotes the movement of the occlusion patterns between the $t$th and $(t+1)$th frames. Since we have assumed that each occluded region has a spatial continuity and moves smoothly across consecutive frames, we define

$$\begin{aligned} p(\mathbf{z}|\boldsymbol{\theta}) &= p(\mathbf{z}^{(1)}) \prod_{t=1}^{T-1} p(\mathbf{z}^{(t+1)}|\mathbf{z}^{(t)}, \boldsymbol{\theta}^{(t)}) &\quad (6)\\ &= p(\mathbf{z}^{(1)}) \prod_{t=1}^{T-1} \frac{1}{Z_t} \exp\{-E(\mathbf{z}^{(t+1)}, \mathbf{z}^{(t)}, \boldsymbol{\theta}^{(t)})\}, &\quad (7) \end{aligned}$$

where $Z_t$ is a normalizing constant and the energy function $E$ is defined by

$$\begin{aligned} &E(\mathbf{z}^{(t+1)}, \mathbf{z}^{(t)}, \boldsymbol{\theta}^{(t)}) \\ &= -J_{\mathrm{self}} \sum_i z_i^{(t+1)} - J_{\mathrm{inner}} \sum_{i \sim j} z_i^{(t+1)} z_j^{(t+1)} - J_{\mathrm{move}} \mathbf{z}^{(t+1)\top} G(\boldsymbol{\theta}^{(t)}) \mathbf{z}^{(t)}, \end{aligned} \tag{8}$$

where $i \sim j$ means "$i$ and $j$ are adjacent pixels." The transition matrix $G(\boldsymbol{\theta}^{(t)})$ is a function of $\boldsymbol{\theta}^{(t)}$ and $G(\boldsymbol{\theta}^{(t)})\mathbf{z}^{(t)}$ represents the predicted occlusion pattern in the $(t+1)$th observation based on the occlusion pattern in the previous frame and the transition amount $\boldsymbol{\theta}^{(t)}$. The constants $J_{\mathrm{self}}$, $J_{\mathrm{inner}} (> 0)$, and $J_{\mathrm{move}}$ $(> 0)$ represent the tendency of occlusion, a degree of correlation of neighbor pixels in the occlusion pattern, and a degree of correlation between $\mathbf{z}^{(t)}$ and $\mathbf{z}^{(t-1)}$ that determines how close the occlusion pattern $\mathbf{z}^{(t+1)}$ should be to the previous one, respectively. Although it is possible to estimate the parameters $J_{\mathrm{self}}$, $J_{\mathrm{inner}}$ , and $J_{\mathrm{move}}$ from the given LR images, in this study they are given manually for focusing on the estimation of HR images.

The initial distribution $p(\mathbf{z}^{(1)})$ is defined as the same as $p(\mathbf{z}^{(t+1)}|\mathbf{z}^{(t)}, \boldsymbol{\theta}^{(t)})$ except the energy function contains only the fist two terms of (8).

### 3.4 Prior for Occlusion Transition

The occlusion transition parameters $\boldsymbol{\theta}$ are regarded as hidden variables that have a Markov property:

$$p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}^{(1)}) \prod_{t=1}^{T-1} \mathcal{N}(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}, (rI)^{-1}), \tag{9}$$

where $r$ is a scalar precision and $I$ is the identity matrix. This prior asserts that the occluder is under the inertial law and likely to move towards the same direction as in the previous frame.

## 4 Approximate Bayesian Learning

Although the mean of the posterior $p(\mathbf{x}|\mathbf{y})$ is necessary for superresolution, the exact evaluation is intractable because we need to integrate out all the hidden variables and this cannot be performed analytically. Then, we employ the variational Bayes (VB) method and the Laplace approximation in order to derive a practical algorithm.

In the VB method, the joint posterior distribution $p(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{y})$ is approximated by another distribution $q(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta})$, which is called a trial distribution. The trial distribution is determined so as to minimize the Kullback-Leibler divergence between the true posterior and the trial distribution

$$D_{\mathrm{KL}}(q||p) \equiv \left\langle \ln \frac{q(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta})}{p(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{y})} \right\rangle_{q(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta})}, \tag{10}$$

where the square brackets $\langle \cdot \rangle_{q(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta})}$ denote an expectation operator with respect to $q(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta})$. The Kullback-Leibler divergence is always nonnegative and equal to zero only when $q = p$.

To make the optimization tractable, we impose a factorization assumption on the trial distribution: $q(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta}) = q(\mathbf{x}) \prod_{t,i} q(z_i^{(t)}) q(\boldsymbol{\theta})$ and the trial distribution is optimized by a coordinate-descent procedure. The coordinate-descent minimization with respect to the factorized trial distribution gives closed-from solutions $q^*(\mathbf{x})$, $q^*(z_i^{(t)})$, $q^*(\boldsymbol{\theta})$.

As we will see later, $q^*(\mathbf{x})$ and $q^*(z_i^{(t)})$ are given by a Gaussian distribution and a Bernoulli distribution, respectively, and the integrals with respect to them are tractable. However, the integral with respect to $q^*(\boldsymbol{\theta})$ still remains intractable because $\ln q^*(\boldsymbol{\theta})$ is a complicated function of $\boldsymbol{\theta}$. To overcome this difficulty, we utilize the Laplace approximation; i.e., $q^*(\boldsymbol{\theta})$ is approximated by a Gaussian distribution, whose mean $\bar{\boldsymbol{\theta}}$ gives the maximum of $q^*(\boldsymbol{\theta})$.

### 4.1 Optimal Trial Distribution

Due to the VB approximation, the optimal trial distribution for $\mathbf{x}$ is found to be a Gaussian distribution

$$q^*(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\mathbf{x}}, \Sigma_{\mathbf{x}}), \tag{11}$$

where $\boldsymbol{\mu}_{\mathbf{x}} = \Sigma_{\mathbf{x}}(\sum_t W^{(t)\top}\langle B(\mathbf{z}^{(t)})\rangle \mathbf{y}^{(t)})$ and $\Sigma_{\mathbf{x}} = (\rho A + \sum_t W^{(t)\top}\langle B(\mathbf{z}^{(t)})\rangle W^{(t)})^{-1}$.

The direct evaluation of the probability distribution $q^*(\mathbf{x})$ is practically hard due to the inversion of the huge matrix $\Sigma_{\mathbf{x}}$. However, we can skip the direct calculation of $\Sigma_{\mathbf{x}}$ because all we have to know is $\boldsymbol{\mu}_{\mathbf{x}}$ for the final output image. Noting that the inverse of $\Sigma_{\mathbf{x}}$ is a sparse matrix, $\boldsymbol{\mu}_{\mathbf{x}}$ is obtained by a conjugate gradient (CG) algorithm [7].

Due to the VB and Laplace approximation, the optimal trial distribution for $\boldsymbol{\theta}$ becomes a Gaussian

$$q^*(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\boldsymbol{\theta}}, \Sigma_{\boldsymbol{\theta}}), \tag{12}$$

where $\boldsymbol{\mu}_{\boldsymbol{\theta}} = \bar{\boldsymbol{\theta}}$ and $\Sigma_{\boldsymbol{\theta}} = \langle \bar{H} \rangle^{-1}$. Here, $\langle \bar{H} \rangle$ is the Hessian of the $\langle \ln q^*(\bar{\boldsymbol{\theta}})\rangle_{q(\mathbf{z})}$.

The optimal trial distribution for $z_i^{(t)}$ is given by the following Bernoulli distribution:

$$q^*(z_i^{(t)}) = \nu_{ti}^{\frac{1}{2}(1+z_i^{(t)})}(1-\nu_{ti})^{\frac{1}{2}(1-z_i^{(t)})}. \tag{13}$$

The parameter $\nu_{ti}$, which is the probability of the occlusion absence $q(z_i^{(t)} = +1)$, is calculated by a logistic sigmoid function $\nu_{ti} = \text{sigmoid}(-2\lambda_{ti})$, where

$$\begin{aligned} \lambda_{ti} = {} & J_{\text{self}} + J_{\text{inner}} \sum_{j \in N(i)} \langle z_j^{(t)} \rangle + J_{\text{move}}[\bar{G}^{(t-1)}\langle \mathbf{z}^{(t-1)}\rangle + \bar{G}^{(t)\top}\langle \mathbf{z}^{(t+1)}\rangle]_i \\ & + \frac{1}{2} J_{\text{move}} \sum_{j,l,k} [\bar{G}^{(t-1)}_{\theta_k\theta_l ij}\langle z_j^{(t-1)}\rangle \Sigma_{\boldsymbol{\theta}kl} + \bar{G}^{(t)}_{\theta_k\theta_l ji}\langle z_j^{(t+1)}\rangle \Sigma_{\boldsymbol{\theta}kl}] \\ & + \frac{1}{4}\left[\ln(\beta_{\text{H}}/\beta_{\text{L}}) - (\beta_{\text{H}} - \beta_{\text{L}})\langle e_{ti}^2 \rangle\right]. \end{aligned} \tag{14}$$

Here $N(i)$ is the set of the neighboring pixels of pixel $i$ and partial derivatives are denoted by $\bar{G}^{(t)}_{\theta_k} = \partial G(\bar{\boldsymbol{\theta}}^{(t)})/\partial\theta_k^{(t)}$ and $\bar{G}^{(t)}_{\theta_l\theta_k} = \partial^2 G(\bar{\boldsymbol{\theta}}^{(t)})/\partial\theta_l^{(t)}\theta_k^{(t)}$. The first term stems from the self connection, the second term from the inner-image connection, and the third and fourth terms from the occlusion movement. The last term is determined by the observations such that the baseline $(\beta_{\mathrm{H}} - \beta_{\mathrm{L}})^{-1} \ln(\beta_{\mathrm{H}}/\beta_{\mathrm{L}})$ is compared with the following expected squared error at the $i$th pixel of the $t$th observed image: $\langle e_{ti}^2 \rangle = (y_i^{(t)} - \mathbf{w}_i^{(t)\top} \boldsymbol{\mu}_{\mathbf{x}})^2 + \mathbf{w}_i^{(t)\top} \Sigma_{\mathbf{x}} \mathbf{w}_i^{(t)}$, where $\mathbf{w}_i^{(t)}$ is the $i$th row of $W^{(t)}$. Here, the first term represents the reconstruction error on the $i$th pixel of the $t$th image, and the second term represents the degree of uncertainty. This equation means that, if the reconstruction error at $i$th pixel is large, then the $i$th pixel is considered to be occluded.

The VB algorithm iteratively updates each component of the trial posterior, $q^*(\mathbf{x})$, $q^*(\mathbf{z})$ and $q^*(\boldsymbol{\theta})$, one by one by fixing the other components, until convergence. The algorithm is terminated when the relative change of $\boldsymbol{\mu}_{\mathbf{x}}$'s norm is smaller than a predetermined threshold $10^{-4}$.

# 5 Experiments

## 5.1 Synthetic Dataset

The proposed algorithm was tested with synthetic data generated by the following procedure. A given original HR image $\mathbf{x}$ (Lena) was first transformed by translational and rotational motions, where the amounts of shift and rotation were randomly drawn from the respective uniform distributions Unif$(-2, 2)$ and Unif$(-4\pi/180, 4\pi/180)$, blurred with a Gaussian PSF with standard deviation 2, then downscaled by a factor of 4 in each direction, and corrupted by Gaussian noise whose variance is determined by the occlusion pattern $\mathbf{z}^{(t)}$. There were fifteen LR images. The occlusion pattern $\mathbf{z}^{(t)}$ is shown in Fig. 1. Noise SNR was set to 10 dB for the pixels where $z_i^{(t)} = -1$ (occluded pixels), while it was set to 40 dB for the pixels where $z_i^{(t)} = +1$ (intact pixels). The occlusion pattern on the first frame was generated by Gibbs sampling from the prior distribution $p(\mathbf{z}^{(1)})$, and the occlusion patterns on the other frames were generated according to $G(\boldsymbol{\theta}^{(t)}$. The values of $\boldsymbol{\theta}$ were drawn from a Gaussian distribution with a uniform mean vector. We fixed the model parameters at hand-tuned values: $\rho = 300$, $\gamma = 0.5$, $\beta_{\mathrm{H}} = 3 \times 10^5$, $\beta_{\mathrm{L}} = 10$.

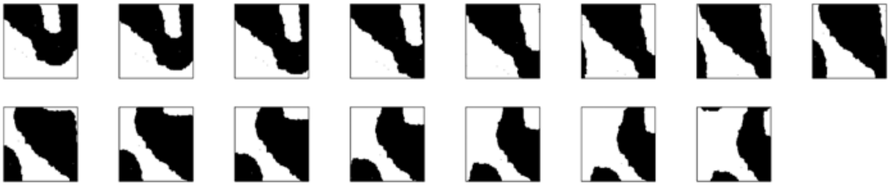
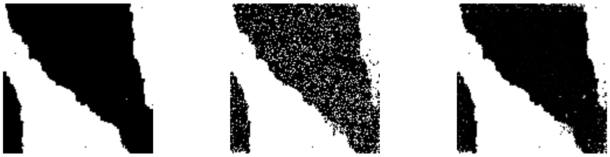
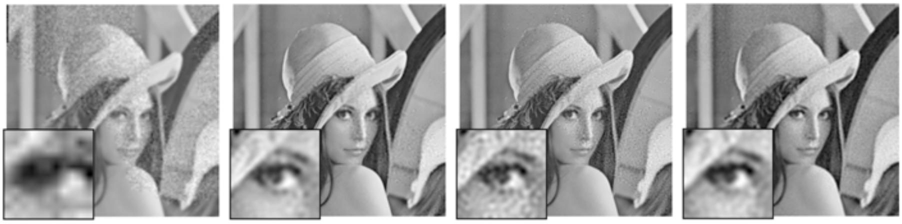


**Fig. 1.** The binary occlusion patterns (black: large noise of 10 dB, white: small noise of 40 dB). The occlusion patterns were shifted according to $\boldsymbol{\theta}$.



(a) True (b) Without move (c) Proposed

**Fig. 2.** True and estimated occlusion patterns



(a) Observation (b) Proposed (c) Without move (d) Uniform

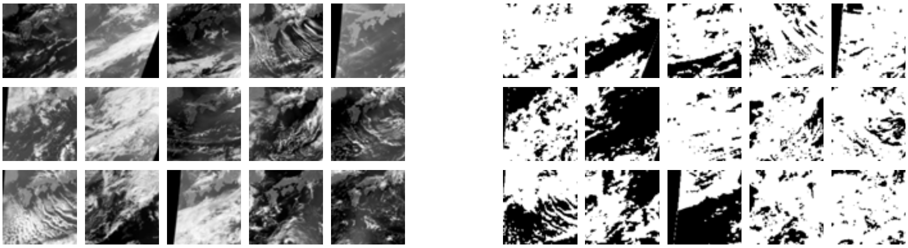
**Fig. 3.** Comparison of estimation results when the series of observed images suffer form the occlusions shown in Fig. 1. (a) One of the fifteen LR images. (b) Proposed (30.69 dB). (c) Without movement (29.13 dB). (d) Assumed uniform noise (28.66 dB).

Fig. 2 shows the true occlusion pattern and its estimates with/without assuming the movement of the occlusion patterns. The estimation results and close-up views of the region around the right eye of Lena are shown in Fig. 3. Fig. 3 shows (a) the first frame of the fifteen LR images, (b) the superresolved image estimated by our method, (c) the estimate without considering the temporal transition of the occlusions, (d) the estimate without assuming occlusions at all. Performance of the algorithm was measured by the peak signal-to-noise ratio (PSNR). As can be seen from the figure, our proposed method achieved the best PSNR (30.69 dB).

### 5.2 Cloud-Contaminated Dataset

The second experiment is to obtain an HR image from a cloud-contaminated NOAA/AVHRR dataset published online by JADAS. We generated a set of LR images synthetically downsampled (by linear scaling of 4 to 1), again using the model with translational and rotational motions and blurred by Gaussinan PSF. The set of the LR images is shown in Fig. 4(a). The model parameters are assumed to be known.

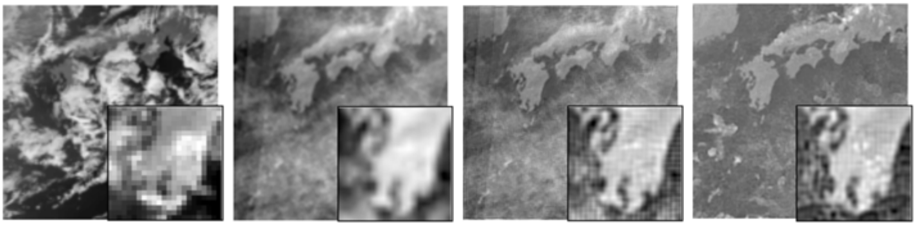
Since the observations are not continuous in time, we do not assume occlusion transition here; the parameter for the occlusion prior $J_{\text{move}}$ was set to 0. We fixed the model parameters at hand-tuned values: $J_{\text{self}} = 0.02, J_{\text{inner}} = 10, \rho = 400, \beta_{\text{H}} = 8 \times 10^3,$ and $\beta_{\text{L}} = 10$.



(a) Observations

(b) Estimated occlusion pattern

**Fig. 4.** Observed images and estimated occlusion patterns



(a) Observation

(b) Median

(c) Uniform

(d) Proposed

**Fig. 5.** Estimation results using the cloud-contaminated images

Fig. 4(b) shows the estimated occlusion patterns. Fig. 5(a) presents an LR image, and Fig. 5(b) shows the observation median. Fig. 5(c) is the superresolved HR image by the simple method without assuming any occlusions. Fig. 5(d), we shows the result by the proposed algorithm. In this experiment, we cannot evaluate PSNR because we do not know the clean HR image. However, it appears that our model gives the best estimation in which the occlusions are removed successfully, and moreover, the quality of the HR image is enhanced. It is expected that the performance will be further improved if sequential observations are used.

## 6 Conclusion

We have proposed an image superresolution method for observations that are contaminated by occlusions. The key factor of our model is the hierarchical likelihood in which the hidden variables are utilized so as to represent the knowledge on the observation process. Uncertainty regarding the HR image, occlusion, and amounts of occlusion movement is effectively considered via Bayesian inference, and our model enables successful reconstruction by reducing the effects of occlusions and accomplishes the improvements in PSNR and the reconstructed HR image.

## References

1. Kanemura, A., Maeda, S., Ishii, S.: Image superresolution under spatially structured noise. In: IEEE Int. Symp. Signal Processing and Information Technology, December 2007, pp. 275–280 (2007)
2. Farsiu, S., Robinson, D., Elad, M., Milanfar, P.: Fast and robust multi-frame super-resolution. IEEE Trans. Image Process 13(10), 1327–1344 (2004)
3. Babacan, D., Molina, R., Katsaggelos, A.: Total variation super resolution using a variational approach. In: IEEE Int. Conf. Image Process, October 2008, pp. 641–644 (2008)
4. Pickup, L.C., Capel, D.P., Roberts, S.J., Zisserman, A.: Bayesian image super-resolution, continued. In: Advances in NIPS. MIT Press, Cambridge (2006)
5. Ivanovski, Z.A., Panovski, L., Karam, L.J.: Robust super-resolution based on pixel-level selectivity. In: Visual Commun. Image Process. Proc. SPIE, vol. 6077, pp. 80–87 (2006)
6. Melgani, F.: Contextual reconstruction of cloud-contaminated multitemporal multispectral images. IEEE Trans. Geosci. Remote Sens 44(2), 442–455 (2006)
7. Golub, G.H., van Loan, C.F.: Matrix Computations, 3rd edn. Johns Hopkins University Press, Baltimore (1996)

# Generating Self-organized Saliency Map Based on Color and Motion

Satoru Morita

Faculty of Engineering, Yamaguchi University

**Abstract.** A computational theory concept of generating saliency maps from feature maps generated from the bottom-up approach using various filters such as a Fourier Transform was discussed. We propose a new method which generates a saliency map by using self-organized filters and not by using general filters such as Fourier transform. We extend the ICA base function estimation to the non-regular positioned photoreceptor cells, which receive the hue image, the saturation image, the current intensity image and the previous intensity image, to get the color and motion information. Our model is expanded so that the filter with the receptive field has a non-uniform arrangement like human with foveated vision. An initial vision model such that a photoreceptor receives color and motion is proposed in this paper. We show the effectiveness of our model by applying this model to real images.

## 1 Introduction

Marr showed that there is a calculation model of the initial vision that extracts a feature from the image from the bottom-up approach[1]. Koch and Ullman proposed the general idea by using various filters to compute the saliency map to show saliency in a two-dimensional plane from the feature map an image generated from the bottom-up approach[2]. Itti and Koch were applied as a calculation model of the form which can be applied to the image analysis[3]. A filter is the feature extraction of the multresolution, which is constructed using a general Fourier Transform and a Rotation Filter. Because these use a general filter, it is necessary for the image to be uniformly arranged. Fourier Transform and Wavelet were used to extract a feature from the image [4]. The information related to the frequency by multiplying a sine wave in the original signal can be obtained with a Fourier Transform. Here, a base function becomes a sine wave. The information related to the frequency by convoluting a mother base function in the original signal can be obtained with Wavelet. They are this technique based on the defined base function. On the other hand, the method of extracting a feature signal by estimating a ICA base was discussed [5]. ICA is an abbreviation of the Independent Component Analysis. An example is not applied for the image arranged non-uniformly though the example is applied for the two-dimensional image and one dimension signal. In this paper, the defined general filter such as Fourier transform isn't used, but the self-organizing filter generated only from the observation of the image is used. Therefore the general

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 28–37, 2009.

idea of the ICA base estimation is introduced in the model which generates a saliency map. The receptive field of the human vision has an arrangement which isn't especially uniform. For example, the density of the receptive field is high in the center and low in the circumference, with a foveated vision. The self-organizing filter is generated from only observing. The receptive field has such arrangement which isn't uniform. Initial vision is modeled based on color and motion which is important for the saliency map. The function of the vision system is mentioned in section 2. The vision requirement is mentioned in section 3. Our model based on the ICA base estimation and the salinecy map according the the vision requirements is mentioned in the section 4. We extend our method to the method applied multiple information such as color and motion. It is actually applied to the image in section 5, and the effectiveness of the initial vision model to extract a remarkable feature based on the ICA base function estimation is shown.

## 2 The Function of the Human Vision System

We survey the function of the human vision system [1] related to this paper in the following. A retina is constructed by layers which are composed of the photoreceptor, the bipolar cell and the ganglion cell. There are rods which react to color and cones which do not react to color in the photoreceptor. The intensity of light changes the membrane potential. A change in the membrane potential in many photoreceptors is integrated, and a bipolar cell transmits it to the gangliocyte. The gangliocyte, which has only an impulse occurrence function in the cell of the intraretinal generates the impulse of the frequency corresponding to the optical strength. As a gangliocyte receives information from many photoreceptors, a gangliocyte reacts to the vision stimulus in the fixed range. A change is caused for the activities of the gangliocyte when stimulus is given to a certain territory. A stimulus influence of gangliocyte in the circumference part is often different from the stimulus influence in the central part.

The receptive cell gets strongly excited when stimulus is given to the central part of the receptive cell, while the excitement is restrained when stimulus is given to a circumference part. This is called the on-center, off-surround form cell. The cell that the influence of the inhibition is taken in the circumference part and the influences of the repression is taken in the central part, and it is called an off-center, on-surround form cell in the reverse. The gangliocyte with the receptive cell reacts on the boundary between light and shade. The fiber of the gangliocyte is connected to the primary visual cortex of the occipital lobe. Two cellular properties of the primary visual cortex are in the following. One is that the form of the receptive cell is rectangle, and another one is that the receptive cell reacts to a stimulus selectively. The cell of the primary visual cortex is classified into simple cell,complex cell, and hypercomplex cell according to that response selectivity. The cell which has the specifical feature selectively is known as the feature extracting cell. In the simple cell, on-response cell and off-response cell which is a rectangle is side by side. Therefore, such a cell reacts to the slit-shaped vision stimulus. A response is poor if the direction, width and position of vision stimulus do not match in the slit-shape. In other words, a simple form cell has selectivity against the direction, width and position of the slit. Selectivity to the direction of the stimulus is specially named an orientation sensitivity. The receptive field of a complex cell is wider than that of a simple cell, and a complex cell reacts to the complex feature selectively. Selectivity against the position of the slit light is poor without a complex cell reacting even if the vision stimulus of the punctate is given to it, too. But, it has strong selectivity to the direction of the slit light and the width. Most of the complex cells react to the stimulus to move in the specifical direction which specially has the stimulus that it is rather moved more strongly than the vision stimulus to stand again. Though a hypercomplex cell looks like a complex cell, it reacts only when a more complex condition is satisfied. Hubel and Wiesel discovered that there was a neuron which reacted when the slit light of the specifical inclination is reflected to the retina in the visual cortex of the cerebral membrane of the cat[6][7]. Such a cell is called a feature extracting cell. The cells discovered by Hubel and Wiesel have multiple kinds, and they responded to the slit light of each inclination. If the cat which didn't have a vision experience grows up in a cage and sees only vertical sticks, the cat does not have the feature extracting cell responding to the horizontal slit light but the feature extracting cell responding to the vertical slit light. Malsburg explained how the neuron which Hubel and Wiesel discovered is arranged using the neural network model including Hebb learning[6][7]. Willshaw and Malsburg showed that structure of the combination of the topological mapping seen in the living body was formed by using the neural network model[11]. Kohonen expanded the interpretation of the neural network model of the topological mapping from the stand point of the information processing[12]. The function of the vision system surveyed is realized in this paper.

## 3 A Requirement for the Vision

{1} The feature which was dependent on the input conditions can be detected.
{2} The input position of the receptive field can be not regular.
{3} Only information that the retina receives is analyzed.
{4} The feature can be detected without depending on the order observing the image.

To realize requirement {1}, we introduce ICA base funtion estimation for generating the saliency map. To realize the requirement {2}, Because input for the ICA base function estimation can be applied to the arrangement which isn't uniform, it is expanded to realize the requirement {3}, Information to receive is restricted to the minimum information which a living body receives. The requirement {4} can be realized by the feature extraction based on ICA.

{5} Motion and the color can be distinguished.
{6} An edge can be distinguished.
{7} A texture can be distinguished.

We make the receptive field to receive the color and motion information to realize the requirement {5}. We make the recetptive field to receives the neighbor information which is needed for the texture analysis and the edge detection at the same time to realize the requirements {6} and {7}. We make receptive field to be arranged spatially.

{8} The various filters which a person's living body has can be generated.

The requirements {8} are discussed after the result of an experiment.

## 4 Saliency Map and Self-organized Filters Based on ICA Base Function Estimation

The method to estimate an ICA base function from the image and the signal was discussed. The self-organization of the feature extracting cell of the color and the motion are modeled based on the ICA base function estimation in this paper. The feature is detected such that the data projected to the subspace is independent and distributed. When the data that

$$\boldsymbol{X_l} = \{x_{l1}, x_{l2}, \cdots, x_{ln}\}$$

is centering, the result data

$$\boldsymbol{Y_l} = \{y_{l1}, y_{l2}, \cdots, y_{ln}\}$$

is defined in the following as

$$\boldsymbol{Y_l} = \boldsymbol{W_l}\boldsymbol{X_l}$$

where

$$\boldsymbol{W_l} = \{w_{l1}, w_{l2}, \cdots, w_{ln}\}$$

The saliency map and the ICA base function are estimated for the $l$th data $X_l$. The independence $J0_l$ of $Y_l$ is evaluated as

$$J_l = \Sigma_{i,j(i \neq j)} E\{y_{li} \cdot y_{lj}\}, \tag{1}$$

where $J_l(l = 1, \cdots, m)$ is the independency evaluation value of the $J$ value of the $m$ individual here.

The uncorrelation is weak type of the independency. In this paper, we use the correlation insted of the independency. The two variables $y_1$ and $y_2$ relate uncorrelation if these coveriance $cov(y_1, y_2) = E\{y_1 y_2\} - E\{y_1\}E\{y_2\}$ equals zero. In this paper, as the mean of $y_1$ and $y_2$ are zero, equation (1) means the correlation. The function $E$ is the expectation of the probability variable.

The variables $\boldsymbol{W_l}$ and $\boldsymbol{X_l}$ are decided as so that $J_l$ may become the smallest here. $W_l$ is the estimated base function. It is solved by the method of steepest descent.

The salience map is defined based on the feature value $Sa_l = max\{y_{l0}, \cdots, y_{ln}\}$ and the feature number $Sb_l = \Sigma_{i=0}^{n} f(y_{li})$. A function $f$ is step function.

A saliency map based on the feature value is generated by using $Sa_l(l = 1, \cdots, m)$. A saliency map based on the number of features is generated by using $Sb_l(l = 1, \cdots, m)$.

It is known that the arrangement of the receptive field is often circle and rectangle. Generally the receptive fields of the biological system are seldom arranged uniformly. The receptive field is arranged according to the probability distribution which is dense in the center and is sparse in the periphery. Because the input of the image is two dimensions, a probability density function is defined in the two dimension as well. If the number of the receptive field is 10, $n$ in $X = \{x_1, x_2, \cdots, x_n\}$ becomes 10. In the case of a cell to react to the motion and the intensity, $n$ becomes $10 \cdot 2 = 20$ when the input which varies in each receptive field is received.

## 5 Generating Saliency Map Related to Color and Motion

The distinction of color is important as the distinction of motion is important. When a color is distinguished, it is mentioned that intensity, saturation and hue are important. The input of the receptive field is intensity, saturation and hue and the intensity of the previous frame to detect features of motion and color. The photoreceptor receives 18 inputs. The figure 1(a)(b)(c) and (d) show the four layers of the photoreceptors. The probability that a receptive field exists in the center is high, and the probability that a receptive field exists in the periphery is low. It was arranged in accordance with this probability distribution. This brain

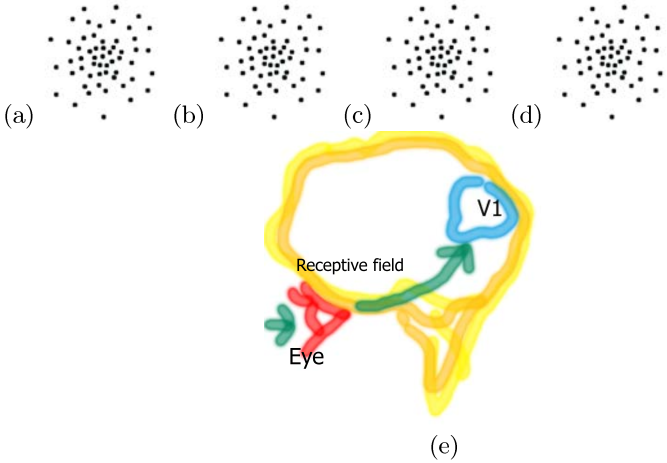


**Fig. 1.** (a)(b)(c)(d) Four layers photoreceptors to detect color and motion. image. (b) A photoreceptor for the current image. (c) The probability distribution of the receptive field of the foveated vision (e)This brain model is the process from eye to the primary visual cortex.

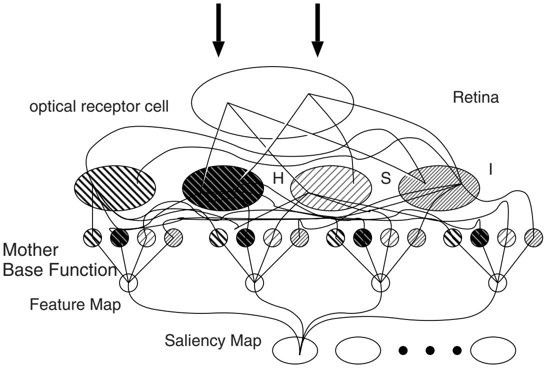


**Fig. 2.** Flow from the photoreceptor which reacts to color and motion to the primary visual cortex

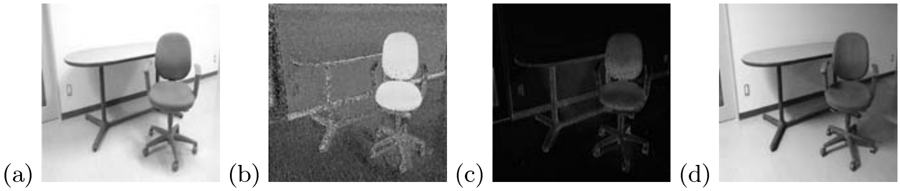


**Fig. 3.** Intensity image, hue image, saturation image and the intensity image of the previous frame

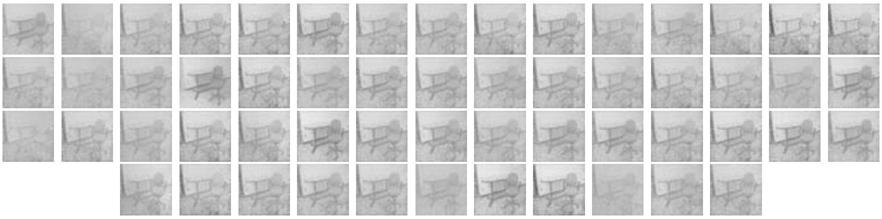


**Fig. 4.** 56 feature images

model between eye and the primary visual cortex as shown in the figure 1(e) is discussed in the paper. Figure 2 shows the flow from the photoreceptor which reacts to color and motion to the primary visual cortex.

- Photoreceptors receive the intensity, the saturation, the hue and the intensity information of the previous frame respectively.
- The features are computed by multiplying the estimated base function in information of the intensity, the hue, the saturation and the intensity of the previous frame.

- Saliency map is computed from the features. When there are two or more base functions, saliency map value is calculated from all the base functions. Two kinds of saliency maps are defined. A saliency map value is defined as the maximum of the feature value in a pixel. Another saliency map value is defined as the number that the feature value is over the fixed threshold value.

There are a chair and a desk in the laboratory room while the position of the lightings changes in figure 3. Figure 3 shows an intensity image, a hue image, a saturation image and the intensity image of the previous frame. The size of the image is 512 × 512. The next viewpoint is defined in the equivalent probability from each pixel in the whole image. The 14 inputs of the receptor receive for the intensity image from the previous frame, the intensity image of the current time, the saturation image and the hue image. There are the rods which react to color and the cones which do not react to color in the photoreceptor. Rods correspond

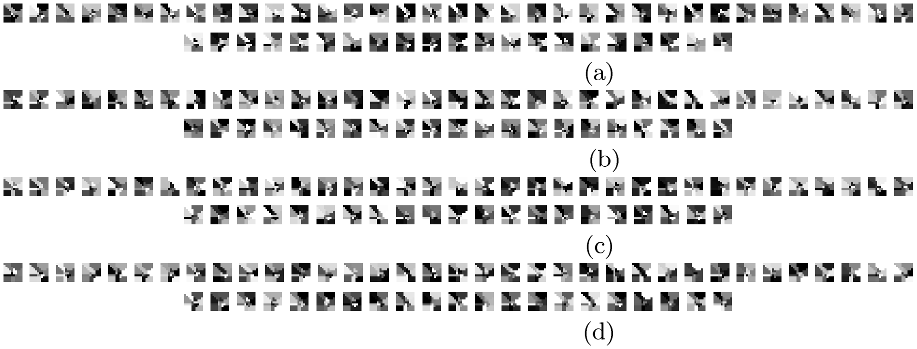


**Fig. 5.** (a) 56 base functions of the receptive cell which receives intensity. (b) 56 base functions of the receptive cell which receives hue. (c) 56 base functions of the receptive cell which receives saturation. (d) 56 base functions of the receptive cell which receives the intensity of the previous frame.

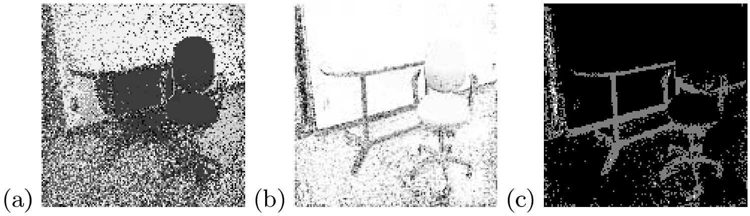


**Fig. 6.** (a) A saliency map based on the feature value for a receptive cell to react to color and motion. (b) A saliency map based on the number of features for a receptive cell to react to color and motion. (c) An event map classified based on features.

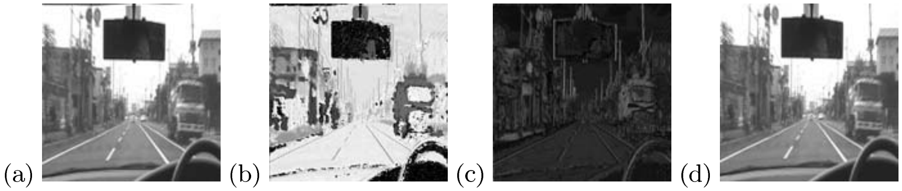


**Fig. 7.** Intensity image, hue image, saturation image and the intensity image of the previous frame

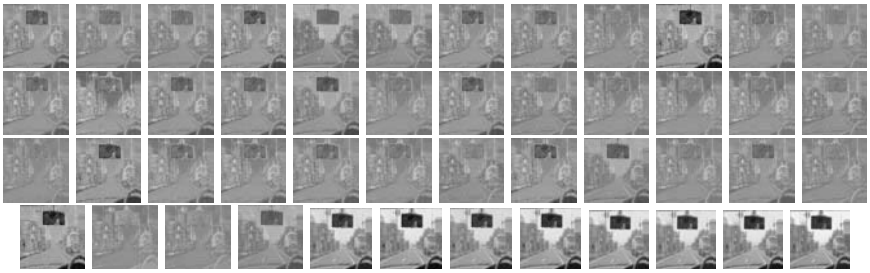


**Fig. 8.** 48 feature images

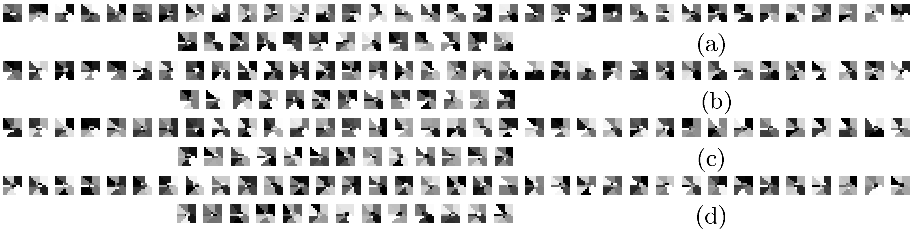


**Fig. 9.** (a) 48 base functions of the receptive cell which receives intensity. (b) 48 base functions of the receptive cell which receives hue. (c) 48 base functions of the receptive cell which receives saturation. (d) 48 base functions of the receptive cell which receives the intensity of the previous frame.

to the receptive cell, which receives the intensity, and cones correspond to the receptive cell which receives the hue and saturation. Figure 4 shows 60 feature images, because $14 \cdot 4 = 56$. Figure 5 (a),(b),(c) and (d) show the 56 base functions of the receptive cell which receives intensity, hue, saturation and the intenisty of the previous frame respectively. Figure 6 is the saliency map computed in this method. Figure 6 (a) and (b) is a saliency map based on the number of features and the maximum feature value respectively. Figure 6 (c) is an event map classified based on features.

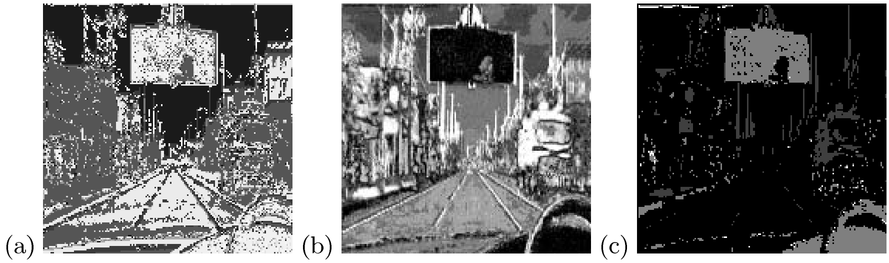


**Fig. 10.** (a) A saliency map based on the feature value for a receptive cell to react to color and motion. (b) A saliency map based on the number of features for a receptive cell to react to color and motion. (c) An event map classified based on features.

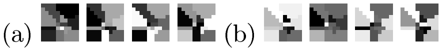


**Fig. 11.** (a)The example of the remarkable base reacts to the intensity in the base of figure 5. (b) The example of the remarkable base reacts to the color in the base of figure 5. Four images are the intensity, hue, saturation and the intensity of the privous time.

First, the image that a driver sees is analized. Figure 7 shows an intensity image, a hue image, a saturation image and the intensity image of the previous frame. The size of the image is $512 \times 512$. The 12 inputs of the receptor receive for the intensity image from the previous frame, the intensity image of the current time, the saturation image and the hue image. Figure 8 shows 48 feature images because $12 \cdot 4 = 48$. Figure 9 (a),(b),(c) and (d) show the 48 base functions of the receptive cell which receives intensity, hue, saturation and the intenisty of the previous frame respectively. A color of a receptive field is often black in the center, and white in the periphery. On the other hand, a color of a receptive field is often white in the center, and black in the periphery. A color is often white only in a certain direction. The various filters which a person's living body has can be generated. The requirement $\{8\}$ is satisfied in the experiment. There are relations between the neighbour images , and it is found that the base function is the reverse relation with a certain base function. These cells are related to the on-center off-surround form cell, off-center on-surround form cell, the orientation selective cell and the motion selective cell. Their cells are seen in the neuron fiber cell and primary vision field. A more remarkable base function becomes clear by increasing the number of the receptive cells. It is meaningful that the self-organizing cell which is common with the actual receptive field is generated. It is understood that the chair color is identified. On the other hand the part that changes from dark to light and the part that changes form light to dark are distinguised from figures 6 (a) and 6 (b). You can find many cells extracting a change in the brightness and color from figure6. The filter equivalent to some living bodies can be confirmed with the optical requirement {8} from a result of observation. As the territory which contains motion and color in the event map is similar to the learned image obtained as a result, the event analysis of task and high level manage based on low level manage can be done. Figure 11 (a) shows the example of the remarkable base reacting to the intensity in the base of figure 5. Figure 11 (b) shows the example of the remarkable base reacting to the color in the base of figure 5. Four images in figure 11 (a) and (b) correspond to the intensity, hue, saturation and the intensity of the privous time.

## 6 Conclusion

We proposed the new brain model of the self-organized saliency map using ICA base estimation to react color and motion. It is found that the several feature extracting cell depending on the driver's view and laboratory view can be generated by applying thie proposed model for the driver's view and the laboratory view.

## References

1. Marr, D.: VISION. W.H. Freeman and Company, New York (1982)
2. Koch, C., Ullman, S.: Shifts in selective visual-attention towards the underlying neural circuitry. Hum. Neurobiol. 4, 219–227 (1985)
3. Itti, L., Koch, C.: Feature combination strategies for saliency-based visual attention systems. J. Electron Imaging 10(1), 161–169 (2001)
4. Mallet, S.G.: A theory for multiresolution signal decomposition: The wavelet representation. IEEE Tran. on PAMI 11, 674–693 (1989)
5. Hyvarinen, A., Hoyer, P.O.: Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspace. Neural Computation 12(7), 1705–1720 (2000)
6. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J. Physiol. 160, 106–154 (1962)
7. Hubel, D.H., Wiesel, T.N.: Receptive fields of cells in striate cortex of very young, visually inexperienced kittens. J. Neurophysiol. 26, 994–1002 (1963)
8. Blakemore, C., Cooper, G.F.: Development of the brain depends on the visual environment. Nature 228, 477–478 (1970)
9. Blakemore, C., Mitchell, D.E.: Environmental modification of the visual cortex and the neural basis of learning and memory. Nature 241, 467–468 (1973)
10. von der Malsburg, C.: Self-organization of orientation sensitive cells in the striate cortex. Kybernetik 14, 85–100 (1973)
11. Willshaw, D.J., von der Malsburg, C.: How patterned neural connections can be set up by self-organization. Proc. R. Soc. Land. B. 194, 431–445 (1976)
12. Kohonen, T.: Self-organized formation of topographically corerct feature maps. Biol. Cybern. 43, 59–69 (1982)
13. Horn, B.K., Schunck, B.G.: Determining optical flow. Artif. Intell. 17, 185–203 (1981)

# Co-occurrence of Intensity and Gradient Features for Object Detection

Akinori Hidaka[1] and Takio Kurita[2,3]

[1] Tokyo Denki University
[2] National Institute of Advanced Industrial Science and Technology (AIST)
[3] University of Tsukuba

**Abstract.** For the purpose of object detection, Haar-Like Features (HLF) proposed by Viola [13][14] are very famous. To classify images, usually HLF and its extensions used only image intensity. However, it is well known that the gradient information of image intensity is very important for the object recognition [2][9]. So in this paper, we propose a feature which uses both intensity and gradient informations. Our feature, called "Co-Occurrence Feature (COF)", can treat the co-occurrence of salient regions in both of intensity domain and gradient domain. We use an extended image set that consists of original (intensity) image and oriented gradient images which are extracted from original images. COF is composed from a pair of arbitrary rectangles on arbitrary image channel in the extended image set. As a result of face/nonface classification experiments, it is confirmed that our feature has good classification performance, especially in the high true positive rate zone of ROC curves, the false detection rate is significantly better than Viola's HLF.

## 1 Introduction

For the purpose of object detection, Haar-like features (HLF) proposed by Viola [13][14] are very famous and many extensions have been studied [4][5][6][7][8][10][12]. HLF is one of local image features and it is composed from several elemental rectangles in an image plane. The feature value of HLF is a difference of average intensity between elemental rectangles on an inputted image. HLF classifies the image by using the feature value as an input of simple classification methods, e.g. decision stumps or naive bayes. Since there are too many HLF in an image plane according to the degree of freedom of elemental rectangles' coordinates, Viola proposed feature selection by Adaboost to choose a small subset of HLF which has powerful classification performance.

The original HLF [13][14] consists of up to four elemental rectangles of the same size which are neighboring each other, and the configuration patterns of rectangles were limited to four types in Fig. 1. Afterwards, the configuration patterns were extended to 14 types by Lienhart [8], and 19 types by Pham [12], to improve the classification power of HLF (Fig. 2). The extended HLFs are composed from rectangles of not always the same size, and it can represent more complex local structures of targets. Hidaka extended HLF to the set of

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 38–46, 2009.

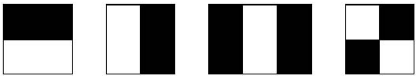


**Fig. 1.** Configurations of elemental rectangles of original HLF

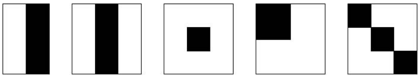


**Fig. 2.** Examples of configurations of extended HLF in [8] or [12]

several arbitrary rectangles in the image plane [5]. The feature can catch a relation between not only the close regions but also distant regions, and it can represent complex structures of targets more flexibly. Mita proposed an extended classifier constructed from multiple HLF [10][11]. The classifier is based on the co-occurrence of HLF and made it possible to construct an effective classifier.

It is well known that the gradient information of image intensity is very important for many image recognition tasks. SIFT descriptor proposed by Lowe [9] and Histograms of Oriented Gradients (HOG) proposed by Dalal [2] achieved strong performance in many computer vision applications, such as matching, retrieval, classification, detection, tracking and motion recognition. These descriptor or feature calculate oriented gradient informations of image intensity. The gradient informations are accumulated into histograms of quantized edge orientations. The histogram is made from a certain local measurement region (called "block") in an image. In the case of HOG, usually a thousands of histograms are calculated from a set of many blocks which are covering the image entirely. Therefore, usually HOG is not computationally efficient. Zhu proposed feature selection for HOG blocks based on Adaboost [3], and they showed it is possible to achieve comparable classification power by using very small number of blocks compared with the case of usual HOG [15].

Perhaps to suppress the computational cost and system complexity, most of described methods employed either intensity or gradient informations. However, the additional information channels of images will lead to improvement of classification performance. So in this paper, we propose a feature which uses both intensity and gradient informations. We consider the extended image set $\{I_0, \cdots, I_B\}$ where $B$ is the number of quantized edge orientation and $\{I_1, \cdots, I_B\}$ is the set of oriented edge images which are made from original image $I_0$. Our feature handles the co-occurrence of rectangles of images, just like [5][10][11]. As a novel point, our feature is composed from a pair of arbitrary rectangles on arbitrary channels in the extended image set. In this paper, we call proposal feature "Co-Ooccurrence Feature (COF)".

In our case, the total number of elemental rectangles increases according to the increasing of image channel. Hidaka showed that it is possible to achieve comparable classification power against Viola's method by using a randomized small subset of HLFs for the feature selection of Adaboost [4]. As well as such method, we use random candidates approach; For the feature selection training by Adaboost, the set of candidate COF is made by selecting a pair of arbitrary rectangles from arbitrary image channel, at random. To maintain a variety of the features, the subset of COF will be regenerated in each iteration of Adaboost.

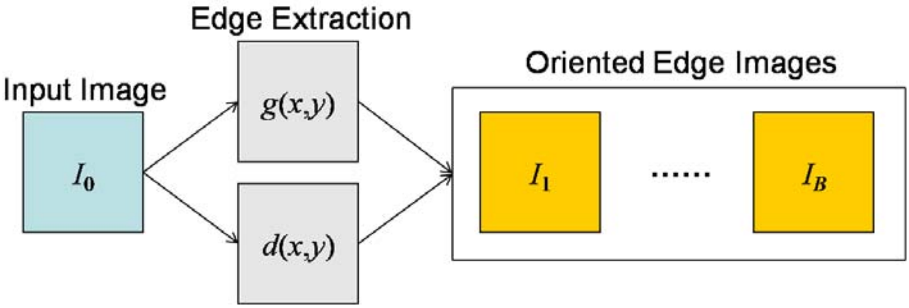


**Fig. 3.** The making process of the extended image set. The process of edge extraction is performed by using Eqs. (1) to (4). Given an image length $W$ and height $H$, $g(x,y)$ and $d(x,y)$ $(1 \leq x \leq W, 1 \leq y \leq H)$ represent the gradient magnitude and gradient orientation of a pixel $(x,y)$, respectively. The oriented edge images are made from $g(x,y)$ and $d(x,y)$ by using Eq. 5. As shown in Eq. 5, $i_b(x,y)$, the pixel value of $b-$th oriented edge image $I_b$, is the weighted gradient magnitude where the weight $\cos|d(x,y) - c_b|$ indicates the degree of similarity between central angle of $b-$th orientation bin and actual gradient orientation of a pixel $(x,y)$.

We valid the performance of COF by face/nonface classification experiments. Our feature has good classification performance compared with Viola's HLF, especially in the zone of high true positive rate of ROC curve, the false detection rate is significantly better.

## 2 Co-Occurrence Feature for Extended Image Set

We propose the novel image feature, called Co-Occurrence Feature (COF), for the extended image set.

### 2.1 Extended Image Set

At first, oriented edge image set $\{I_1, \cdots, I_B\}$ is made from original image $I_0$ ($B$ is the number of bins of the edge orientation to be quantized). The central angle of $b-$th orientation bin is obtained as $c_b = (b-1) * \frac{180}{B}$. To make each edge image $I_b$, every pixels $i(x,y)$ of $I_0$ is filtered by $3 \times 3$ sobel filter, and then the edge magnitude $g(x,y)$ and edge direction $d(x,y)$ are calculated as follows:

$$g_v(x,y) = \sum_{k=-1}^{+1} 2^{1-|k|}(i(x+1,y+k) - i(x-1,y+k)), \tag{1}$$

$$g_h(x,y) = \sum_{k=-1}^{+1} 2^{1-|k|}(i(x+k,y+1) - i(x+k,y-1)), \tag{2}$$

$$g(x,y) = \sqrt{g_v^2(x,y) + g_h^2(x,y)}, \tag{3}$$

$$d(x,y) = \arctan \frac{g_h}{g_v}. \tag{4}$$

The pixel value $i_b(x,y)$ of $b-$th oriented edge image $I_b$ is calculated as follows:

$$i_b(x,y) = g(x,y) \cos |d(x,y) - c_b|. \tag{5}$$

As a result, the extended image set $\{I_0, I_1, \cdots, I_B\}$ is obtained.

### 2.2 COF on Extended Image Set

In our method, feature extraction is performed on the extended image set $\{I_0, I_1, \cdots, I_B\}$. A elemental rectangle $r_b$ on $I_b$ is defined as follows:

$$r_b = (x_s, y_s, x_e, y_e) = \{(x,y) \in I_b | x_s \leq x \leq x_e, y_s \leq y \leq y_e\} \tag{6}$$

where $(x_s, y_s)$ and $(x_e, y_e)$ are the diagonal apexes of the rectangle. A COF is composed from a set of $R$ elemental rectangles $\{r_{b_p}^p\}_{p=1}^R$. Fig. 4 illustrates the difference between HLF and COF.

The feature value of COF is based on the average pixel values $\{m(r_{b_p}^p)\}_{p=1}^R$ on the elemental rectangles $\{r_{b_p}^p\}_{p=1}^R$. The average pixel value on a rectangle $r$ can be calculated as follows:

$$m(r_b) = \iint_{r_b} v(x,y) dx dy \tag{7}$$

where $v(x,y)$ is $i(x,y)$ if $b = 0$ and $i_b(x,y)$ in Eq. (5) if $b > 0$.

HLF employed the difference of average intensity of positive (white) rectangles and negative (black) rectangles, as a feature value. Therefore HLF has

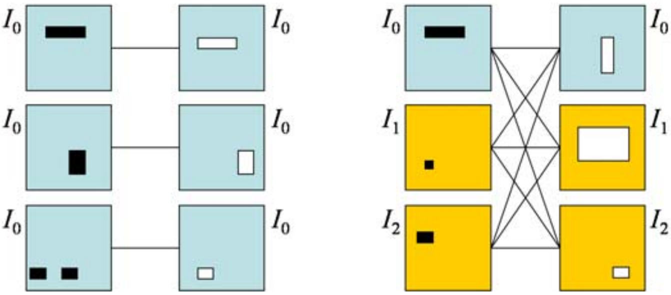


**Fig. 4.** The difference of HLF and COF. (Left) HLF is composed from only the combinations of the same size and neighboring rectangles on the intensity channel. (Right) On the other hand, COF is composed from arbitrary rectangles in arbitrary (i.e., intensity and oriented edge) channels.

robustness about additive illumination changes. On the contrary, in our case, the feature value will be made from different information channels (i.e. intensity and oriented edge). It is hard to consider that a robustness for additive change of intensity image and oriented edge image is useful in a real applications. So we shoud not adhere about the difference of the averages.

In this paper, we make the feature value of COF by two simple ways: to take the "sum" and the "difference" of the average pixel values of elemental rectangles. In the feature selection training, the classifiers based on each feature type are independently trained, and the most superior one is adopted.

In the case of "difference" type, a combinatorial problem of elemental rectangles is caused. However, because of the computational complexity problem indicated in Sec. 3.1, we set the number of elemental rectangles $R = 2$. Therefore the combinatorial problem is not occurred in this paper.

The feature values of COF $\{r^1_{b_1}, r^2_{b_2}\}$ for an image $I$ is represented as follows:

$$f^+(I) = f^+(\{I_b\}^B_{b=0}) = m(r^1_{b_1}) + m(r^2_{b_2}) \tag{8}$$

$$f^-(I) = f^-(\{I_b\}^B_{b=0}) = m(r^1_{b_1}) - m(r^2_{b_2}) \tag{9}$$

The classification function of COF for an image $I$ is written as follows:

$$h(p, \theta, f(I)) = \begin{cases} 1 \text{ if } p(f(I) - \theta) \geq 0 \\ 0 \text{ otherwise} \end{cases} \tag{10}$$

where $p \in \{1, -1\}$ and $\theta \in R$ are parameters determined by training (Fig. 6).

## 3 Adaboost for COF Selection

Adaboost [3] is the ensemble learning method that trains multiple base-classifiers and assembles them to create a more powerful classifier. Viola used HLF as base-classifiers for the face detection [13][14]. The algorithm of Viola's Adaboost is composed from two phases as follows:

1. Weak training: optimizes all HLF independently,
2. Feature selectioin: selects the best one from those optimized HLF.

Therefore the training time of this algorithm depends on the number of HLF.

The training algorithm for COF is described in the following sections.

### 3.1 Random Candidates Approach for COF Selection

In $W_b \times H_b$ pixels image, there are $P_b = W_b(W_b + 1) \times H_b(H_b + 1)/4$ elemental rectangles. Therefore the total number of elemental rectangles on the extended image set is $P = P_0 + P_1 + \cdots + P_B$. As described in Sec. 2.2, COF is composed from $R$ arbitrary elemental rectangles. So the total number of COF is $K = P^R$. For example $B = 2$, $R = 2$ and $W_0 = H_0 = 19$, the case that prameters about training condition are pretty small, $K$ becomes about 7 billion. This is 10,000 times larger than the number of Viola's HLF, and their feature

```
– Input $N$ labeld samples $\{I_i, y_i\}_{i=1}^{N}$ where $I_i$ is a sample and $y_i$ is a label.
    • $y_i = 1 \Leftrightarrow I_i$ is a positive sample, $y_i = 0 \Leftrightarrow I_i$ is a negative sample.
    • $w_i > 0$, $\sum_{i=1}^{N} w_i = 1$.
– Initialize samples weights: if $y_i = 1$ then $w_i = \frac{1}{2p}$, otherwise $w_i = \frac{1}{2q}$.
  ($p, q$: # of positive samples and negative samples, respectively)
– for $t = 1, \cdots, T$
    • Normalize weights: $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{i=1}^{N} w_{t,i}}$.
    • Make the subset of COF at random: $\{h_1, h_2, \cdots, h_{10000}\}$.
    • for $c = 1, \cdots, 10000$
        ∗ Perform weak-training to COF $h_c(p, \theta, f^+)$.
        ∗ Evaluate weighted error rate $we(h_c(p, \theta, f^+))$ of optimized $h_c(p, \theta, f^+)$.
        ∗ Optimize COF $h_c(p, \theta, f^-)$.
        ∗ Evaluate weighted error rate $we(h_c(p, \theta, f^-))$ of optimized $h_c(p, \theta, f^-)$.
        ∗ Set $(h_c, we_c) = (h_c(p, \theta, f^+), we_c^+)$;
          If $we^- < we^+$ then set $(h_c, we_c) = (h_c(p, \theta, f^-), we_c^-)$.
    • Select the $t-$th base-classifier: $b_t = \arg\min_{h_c} we(h_c)$, $e_t = \min_{h_c} we(h_c)$.
    • Compute $\alpha_t = \log((1 - e_t)/e_t)$.
    • Update samples weights: $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot \delta(y_i - b_t(I_i))]$ where
      $\delta(x) = 1$ if $x = 0$, or $\delta(x) = 0$ otherwise.
– Final classification function is
  $H(I) = \begin{cases} 1 \text{ if } \sum_{t=1}^{T} \alpha_t b_t(I) \geq \Theta \sum_{t=1}^{T} \alpha_t. \\ 0 \text{ otherwise.} \end{cases}$ ($\Theta$: Threshold)
```

**Fig. 5.** Adaboost for COF selection. The weak-training algorithm is shown in Fig. 6.

```
– Input COF $h(p, \theta, f)$.
– Input $N$ labeld and weighted samples $\{I_i, w_i, y_i\}_{i=1}^{N}$ where $I_i$, $w_i$ and $y_i$ is a sample,
  a weight and a label.
    • $y_i = 1 \Leftrightarrow I_i$ is a positive sample, $y_i = 0 \Leftrightarrow I_i$ is a negative sample.
    • $w_i > 0$, $\sum_{i=1}^{N} w_i = 1$.
– Let $w^+$ and $w^-$ be sum of weights of positive and negative samples, respectively.
– Extract feature value $f_i = f(I_i)$ from every samples.
– Make up-sorted array $\{g_j\}_{j=1}^{N}$ from $\{f_i\}_{i=1}^{N}$.
– Set $\theta = 0$, $e^+ = w^-$, $e^- = w^+$, $\theta_{best}^+ = \theta_{best}^- = 0$, $e_{best}^+ = e_{best}^- = 1$.
– For $j = 1, \cdots, N - 1$
    • Set $\theta = \frac{1}{2}(g_j + g_{j+1})$.
    • If $y_j = 1$, then
        ∗ Set $e^+ \leftarrow e^+ + w_j$ and $e^- \leftarrow e^- - w_j$.
        ∗ If $e^- < e_{best}^-$ then set $e_{best}^- \leftarrow e^-$ and $\theta_{best}^- \leftarrow \theta$.
    • Else if $y_j = 0$, then
        ∗ Set $e^+ \leftarrow e^+ - w_j$ and $e^- \leftarrow e^- + w_j$.
        ∗ If $e^+ < e_{best}^+$ then set $e_{best}^+ \leftarrow e^+$ and $\theta_{best}^+ \leftarrow \theta$.
– If $e_{best}^+ < e_{best}^-$ then set $p = +1$ and $\theta = \theta_{best}^+$, else set $p = -1$ and $\theta = \theta_{best}^-$.
```

**Fig. 6.** The algorithm of weak-training for COF

selection algorithm can not handle such the huge classifier candidates. Therefore, we use random candidate approach like in [4]. In our experiment, 10,000 features (HLF for Viola's method and COF for our method) are generated at random as the candidate of a base-classifier. After the feature optimization, the best feature in those candidates is selected as the base-classifier. The random candidates are regenerated in each iteration of Adaboost to keep the variaty of features.

Figs. 5, 6 show the algorithm of Adaboost by random candidates approach.

## 4 Experiments

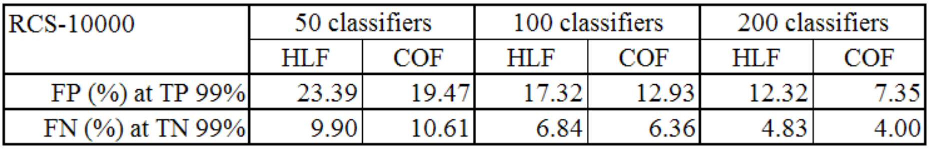
In this paper, we used MIT CBCL face database [1] for our experiments. The database consists of $2,901$ facial and $28,121$ background $19 \times 19$ pixel images. We divided all images into training and test set; $2,000$ face and $5,000$ background images are used for training, and the remaining images are used for testing. Five pairs of training and test sets were generated at random.

We used four types HLF illustrated in Fig. 1 and then the total number of HLF becomes about $53,000$. The parameters of COF are fixed to $B = 2$ and $R = 2$. There are about $83,000$ elemental rectangles in the extended image set, so we can use about 7 billion $(83,000^2)$ COF potentially.

In each iteration of Adaboost for HLF/COF training, the set of 10,000 HLF or COF as the base-classifier candidates is generated at random. The iteration steps of Adaboost (equivalent to the number of the base-classifiers) is up to 200.

Fig. 7 shows the false positive rate (FP) and false negative (FN) rate when true positive rate (TP) equals 99% and true negative rate (TN) equals 99%. When TN = 99%, HLF and COF have comparable results. However, when TP = 99%, the results of COF are significantly better than the results of HLF. In the case of 200 classifiers, FP rate of COF is improved 5% against HLF.

Fig. 8 shows the ROC curves for the five test sets created by using 200 HLF or COF. It can be seen that there are no significant difference between the results of HLF and COF when TP is smaller than 0.95. On the contrary, especially when TP is larger than 0.99, the results of COF are significantly better.



| RCS-10000 | 50 classifiers | | 100 classifiers | | 200 classifiers | |
|---|---|---|---|---|---|---|
| | HLF | COF | HLF | COF | HLF | COF |
| FP (%) at TP 99% | 23.39 | 19.47 | 17.32 | 12.93 | 12.32 | 7.35 |
| FN (%) at TN 99% | 9.90 | 10.61 | 6.84 | 6.36 | 4.83 | 4.00 |

**Fig. 7.** Average error rates (%) for negative (positive) samples at TP (TN) = 99%

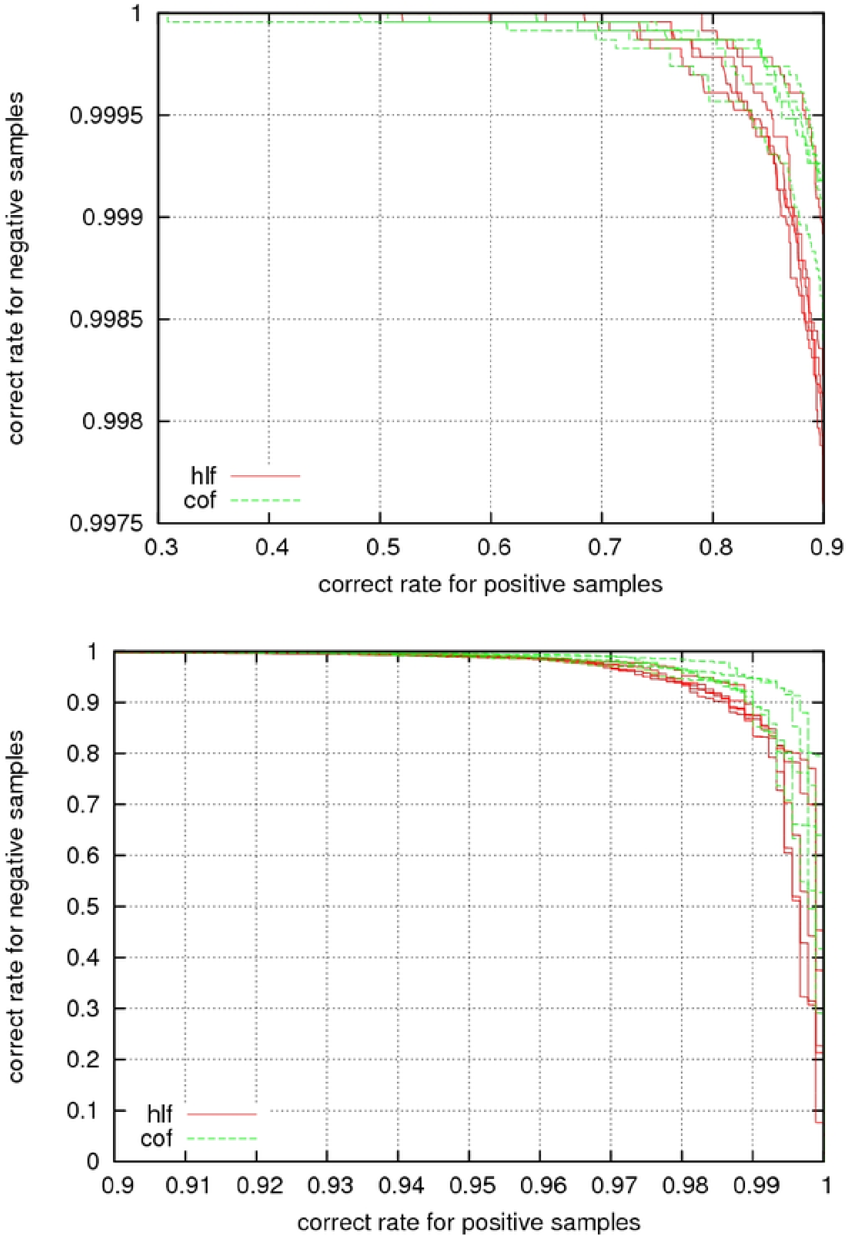


**Fig. 8.** ROC curves for five test sets. Red and green curves show the results of HLF and COF, respectively. Both methods used 200 weak classifiers. Horizontal axis shows the true positive (TP) rate and vertical axis shows the true negative (TN) rate. Top and bottom figures show different TP zones (from 0.3 to 0.9 and from 0.9 to 1.0).

## 5 Conclusions

In this paper, we proposed a new image feature which uses both intensity and gradient informations and can treat co-occurrence of salient regions in both of the information domains. In our experiments, we showed that the more powerful features than the best HLF are certainly included in our 7 billion of COF. We also showed that the feature selection based on random candidates manner can efficiently find such powerful features from those extremely abundant candidates. As a result, the classification accuracy is improved compared with Viola's HLF.

## References

1. MIT Center For Biological and Computation Learning. CBCL Face Database #1, http://www.ai.mit.edu/projects/cbcl
2. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: Proc. of CVPR 2005, pp. 886–893 (2005)
3. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. J. of Computer and System Sciences 55(1), 119–139 (1997)
4. Hidaka, A., Kurita, T.: Fast Training Algorithm by Particle Swarm Optimization and Random Candidate Selection for Rectangular Feature Based Boosted Detector. In: Proc. of IJCNN 2008 on WCCI 2008, pp. 1164–1170 (2008)
5. Hidaka, A., Kurita, T.: Non-Neighboring Rectangular Feature Selection using Particle Swarm Optimization. In: Proc. of ICPR 2008 (December 2008)
6. Huang, C., Ai, H., Li, Y., Lao, S.: High-performance rotation invariant multiview face detection. IEEE Trans. on PAMI 29, 671–686 (2007)
7. Li, S.Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H., Shum, H.: Statistical learning of multi-view face detection. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 67–81. Springer, Heidelberg (2002)
8. Lienhart, R., Maydt, J.: An Extended Set of Haar-like Features for Rapid Object Detection. In: Proc. of IEEE ICIP, vol. 1, pp. 900–903 (2002)
9. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. of Computer Vision 60(2), 91–110 (2004)
10. Mita, T., Kaneko, T., Hori, O.: Proc. of ICCV 2005, pp. 1619–1626 (2005)
11. Mita, T., Kaneko, T., Stenger, B., Hori, O.: Discriminative Feature Co-occurrence Selection for Object Detection. IEEE Trans. PAMI 30(7), 1257–1269 (2008)
12. Pham, M., Cham, T.: Fast training and selection of Haar features using statistics in boosting-based face detection. In: Proc. of ICCV 2007, pp. 1–7 (2007)
13. Viola, P., Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. In: Proc. of CVPR 2001, December 2001, vol. 1, pp. 511–518 (2001)
14. Viola, P., Jones, M.: Robust real-time face detection. Int. J. of Computer Vision 57(2), 137–154 (2004)
15. Zhu, Q., Avidan, S., Yeh, M., Cheng, K.: Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In: CVPR 2006, vol. 2, pp. 1491–1498 (2006)

# Adaptive Sensor-Driven Neural Control for Learning in Walking Machines

Poramate Manoonpong and Florentin Wörgötter

Bernstein Center for Computational Neuroscience (BCCN)
University of Göttingen, D-37073 Göttingen, Germany
{poramate,worgott}@nld.ds.mpg.de
http://www.nld.ds.mpg.de/~poramate/

**Abstract.** Wild rodents learn the danger-predicting meaning of predator bird calls through the paring of cues which are an aversive stimulus (immediate danger signal or unconditioned stimulus, US) and the acoustic stimulus (predator signal or conditioned stimulus, CS). This learning is a form of pavlovian conditioning. In analogy, in this article a setup is described where adaptive sensor-driven neural control is used to simulate biologically-inspired acoustic predator-recognition learning for a safe escape on a six-legged walking machine. As a result, the controller allows the walking machine to learn the association of a predictive acoustic signal (predator signal, CS) and a reflex infrared signal (immediate danger signal, US). Such that after learning the machine performs fast walking behavior when "hearing" an approaching predator from behind leading to safely escape from the attack.

**Keywords:** Learned predator recognition, Associative learning, Legged robots, Sound, Reflexes, Prediction, Central pattern generators.

## 1 Introduction

Animals can effectively avoid dangerous situations including an attack of their predators. Evidence from physiological and ethological studies shows that some prey animals require experience to adaptively respond to predation (learned predator recognition) instead of an innate recognition. For example, wild rodents learn the danger-predicting meaning of predator bird calls through a temporal association of cues which are an aversive stimulus (unconditioned stimulus, US) and the acoustic stimulus (conditioned stimulus, CS) [1]. This mode of learning is a form of classical conditioning which has been discovered by Ivan Pavlov [2]. Generally, it involves presentations of a neutral stimulus (CS) along with a stimulus of some significance (US). Once these two stimuli become associated, animals begin to perform a behavioral response to the CS instead of the US. Inspired by these findings, we simulate here such predator recognition learning for a safe escape on a walking machine. This learned behavior together with the escape response is controlled by adaptive sensor-driven neural control using a correlation based differential Hebbian learning rule. The work presented here extends our previous works [3], [4] by integrating this learning mechanism into the

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 47–55, 2009.

original modular neural control [3], [4] leading to the adaptive behavior. However, the main purpose of this article is not only to demonstrate the biologically-inspired learning on the walking machine (adaptive behavior) but also to show that the adaptive sensor-driven neural control can be a powerful technique to solve sensorimotor coordination problems of many degrees-of-freedom systems and to effectively provide an online learning capability to the systems.

The following section describes the technical specification of the walking machine platform. Section 3 explains the adaptive sensor-driven neural control for the acoustic predator-recognition learning and escape response. Experiments and results are discussed in Section 4. Conclusion and an outlook on future research are given in the last section.

## 2 The Walking Machine Platform AMOS

The six-legged walking machine AMOS [4], [5] is a biologically-inspired hardware platform for studying the coordination of many degrees of freedom, for performing experiments with neural controllers and learning and for the development of artificial perception-action systems employing embodied control techniques. It consists of a two-part body connected by one active backbone-joint (BJ), at which six identical legs and one active tail are attached. Each leg has three joints that are controlled by servomotors: the thoraco-coxal (TC-) joint enables forward (+) and backward (−) movements, the coxa-trochanteral (CTr-) joint enables elevation (+) and depression (−) of the leg, and the femur-tibia (FTi-) joint enables extension (+) and flexion (−) of the tibia (see [5] for the leg configuration of AMOS). Each tibia segment has a spring damped compliant element to absorb impact force during walking. All in all, this machine has 20 sensors: six foot contact ($FC_{1,...,6}$) sensors, seven infrared ($IR_{1,...,7}$) sensors, two light dependent resistor ($LDR_{1,2}$) sensors, one gyro (GR) sensor, one inclinometer (IM) sensor, one upside-down detector (UD) sensor, one current sensor (I) and one auditory-wind detector (AW) sensor (see [5] and also http://www.nld.ds.mpg.de/∼poramate/ICONIP09/AMOSLearning.mpg for the location of all sensors on AMOS). Here the rear $IR_7$ and AW sensors installed at the tail are used for learning experiments of acoustic predator recognition which is the main contribution of this article while the use of other sensors in sensor-driven behavioral applications can be found in [4], [5]. The control of this walking machine is programmed into a personal digital assistant (PDA) with the update frequency of $\approx$ 14 Hz (see [4], [5] for more details of AMOS).

## 3 Adaptive Sensor-Driven Neural Control

The adaptive sensor-driven neural control (Fig. 1) generally generates reactive and adaptive behaviors. The controller based on a modular structure is formed by three main components: a neural preprocessing unit, a modular neural control unit, and a neural learning unit. The neural preprocessing unit filters sensory noise and shapes sensory data to drive corresponding reactive and adaptive behaviors. The modular neural control unit, on the other hand, is used for locomotion generation of the walking machine. It coordinates leg movements, regulates walking speed, and creates omnidirectional walking. Additionally the neural learning unit allows the walking machine to perform adaptive behavior; i.e., it can learn to respond to a conditioned stimulus, e.g., an acoustic signal which is the main focus of this study. All neurons of the adaptive control are modelled as standard additive non-spiking neurons. Their activity develops according to $a_i(t+1) = \sum_{j=1}^{n} W_{ij}\sigma(a_j(t)) + \Theta_i;\ i = 1, \ldots, n$, where $n$ denotes the number of units, $\Theta_i$ represents a fixed internal bias term together with a stationary input to neuron $i$, and $W_{ij}$ the synaptic strength of the connection from neuron $j$ to neuron $i$. The output of the neurons in the neural preprocessing and modular neural control units is given by the standard sigmoid transfer function $\sigma(a_i) = (1 + e^{-a_i})^{-1}$ and the hyperbolic tangent (tanh) transfer function $\sigma(a_i) = \tanh(a_i)$, respectively, while in the neural learning unit is governed by a linear transfer function. Input units are linearly mapped onto the interval [0, 1] for all neurons in the neural preprocessing and learning units and [−1, 1] for those in the modular neural control unit.

### 3.1 Neural Preprocessing of Sensory Signals

In order to simulate the acoustic predator-recognition learning for escaping a close predator attack from behind (conditioned response) on the walking machine, we use the AW sensor [5] for detecting distant acoustic signals (predator signal) and the rear $\mathrm{IR}_7$ sensor for perceiving too near, hence potentially dangerous, approaching objects from behind. Here we use portable speakers [5] implying a predator and the 300 Hz acoustic signal referring to a predator signal. The speakers are manually moved for creating the predatory attack situation (see http://www.nld.ds.mpg.de/∼poramate/ICONIP09/AMOSLearning.mpg).

The raw acoustic and infrared signals coming from the AW and $\mathrm{IR}_7$ sensors, respectively, require preprocessors for eliminating the sensory noise as well as shaping the sensory data. To do so, we utilize the dynamical properties, i.e., hysteresis effects, of a recurrent neuron to construct the preprocessing network of the acoustic signal and the same one is also applied to the infrared signal. Each network consists of a series of single recurrent neurons $H1, 2$. Their output is combined at a threshold output neuron $O$ providing a binary output [0, 1] to the neural learning circuit. The complete neural preprocessing unit is shown in Fig. 1(a) (see [5] for more details). It has a capability to eliminate unwanted noise, shape the sensory data, and also prolong the activation time of the sensory signals. The prolongation of the signals are required in order to obtain the appropriate sensory correlation for a learning mechanism and to allow the walking machine to effectively escape from the predator attack. That is, the action shall persevere for longer than the stimulus itself to ensure a safe escape.

### 3.2 A Neural Learning Circuit

The preprocessed acoustic and infrared signals described in the previous section are fed into the neural learning circuit to modify a plastic synapse during learning. The learning goal in this study is to enable the walking machine to learn to

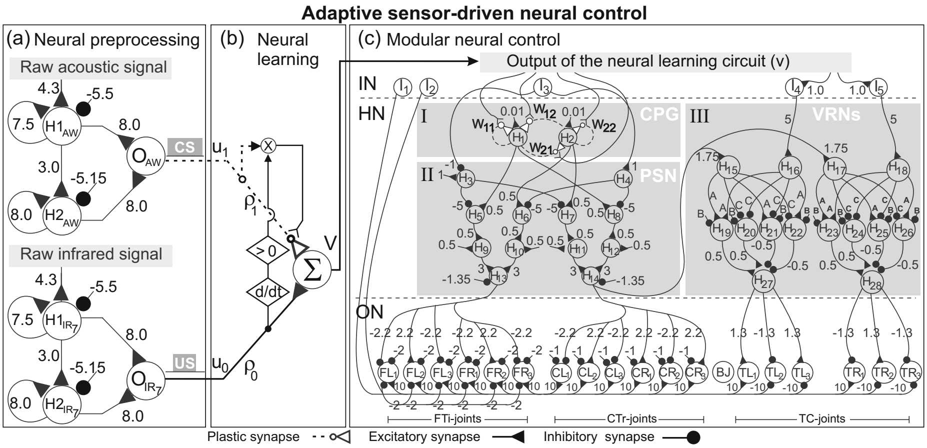


**Fig. 1.** (a) The neural preprocessing unit consisting of two identical networks with appropriate weights (see [5] for more details of neural parameter design and neurodynamics, i.e., hysteresis effects). These networks are applied to preprocess the acoustic and infrared signals. Their output is fed to the neural learning unit. $H1, 2_{AW}$ and $H1, 2_{IR_7}$ = hidden neurons of the AW and IR$_7$ sensor networks with the standard sigmoid transfer function, $O_{AW}$ and $O_{IR_7}$ = output neuron of the AW and IR$_7$ sensor networks with the threshold transfer function, CS = conditioned stimulus, US = unconditioned stimulus. (b) The neural learning circuit where its output activates a fast walking behavior through modular neural control. $\rho_0$ = synapse of the US input set to 1.0., $\rho_1$ = plastic synapse of the CS input, $u_{0,1}$ = inputs to a learner neuron, $v$ = output of the learner neuron. (c) The modular neural control consisting of input (IN), hidden (HN), and output (ON) neuron groups. Input neurons $I$ are the neurons used to control walking direction ($I_{2,...,5}$) and to trigger the self-protective reflex ($I_1$). Hidden neurons $H$ are divided into three modules (CPG, PSN, and VRNs (see [4] for details)). Output neurons ($TR$, $TL$, $CR$, $CL$, $FR$, $FL$) directly command the position of servo motors. The location of the motor neurons on the walking machine AMOS is shown in [5]. BJ = backbone joint, TR(L) = TC-joints of right (left) legs, CR(L) = CTr-joints of right (left) legs, FR(L) = FTi-joints of right (left) legs. All connection strengths together with bias terms are indicated by numbers except those of the VRNs given by A = 1.7246, B = −2.48285, C = −1.7246. $W_{11}$, $W_{22}$, $W_{12}$, and $W_{21}$ are modifiable synapses (see Sect. 3.3).

recognize its acoustic predator signal (300 Hz sound) and then perform a fast walking behavior as soon as it detects the signal. As a consequence, it can avoid the close attack of the predator. To achieve this we apply a correlation based differential Hebbian learning rule [6]. It correlates two kinds of input signals: one is a predictive signal or an early input (conditioned stimulus (CS), see Fig. 1(b)) and the other is a reflex signal or a later input (unconditioned stimulus (US), see Fig. 1(b)). The acoustic signal from the AW sensor is used as the predictive signal while the infrared signal from the IR$_7$ sensor serves as the reflex one. Both sensory signals are provided to the learning mechanism as shown in Fig. 1(b). At the beginning, the connection between the predictive acoustic signal and a learner neuron converges with zero strengths ($\rho_1$, dashed line in Fig. 1(b)). In this situation, the fast walking behavior will be controlled only by the reflex infrared signal. That is, the machine generally walks with normal walking speed. Detecting the acoustic signal and afterwards the objects (portable speakers) from behind here implies the attack of the predator. This situation leads to the correlation between the predictive and reflex signals, such that the modifiable synapse $\rho_1$, which connects the predictive signal with the learner neuron, will grow. Consequently, after 2-3 attacks (depending signal correlation) during the learning phase, the fast walking behavior will finally be driven by the predictive signal instead. The used learning algorithm has the property that learning will stop when the reflex signal is zero [6]; i.e., when the portable speakers (sound source) does not get too near to the walking machine. As the weight is stored, after learning the next time the acoustic signal is detected the fast walking behavior will immediately be activated such that triggering an earlier escape reaction. Eventually the machine will return to its normal walking speed when the acoustic stimulus disappears implying that it is far enough from the sound source or the predator.

*Learning Algorithm*: The correlation based differential Hebbian learning rule (ICO-learning [6]) for the weight change of $\rho_1$ is given by $d\rho_1/dt = \mu u_1 du_0/dt$, where $\mu$ is the learning rate which will define how fast a system can learn, e.g., 0.35. One could consider $\mu$ as the susceptibility for a synaptic change, which in a biological agent will be defined by its evolutionary development, which determines the agent's ability to learn a certain task. How and if these values could also be influenced (possibly by mechanisms of meta-plasticity), changing learning susceptibility, goes beyond the scope of this article. In this learning rule, only the plastic synapse $\rho_1$ is allowed to change while the synapse of the reflex input $\rho_0$ is set to a positive value, e.g., 1.0. Note that here the weight change takes place only at a positive derivative otherwise it remains unchanged. Formally we have $v = \rho_0 u_0 + \rho_1 u_1$ as the learner neuron output driven by inputs $u_{0,1}$. Here we set $u_0$ and $u_1$ to the infrared signal of the IR$_7$ sensor (US) and the 300 Hz acoustic signal of the AW sensor (CS), respectively.

### 3.3 Modular Neural Control

Actions or walking behaviors of the machine are generated through modular neural control. This modular controller consists of three subordinate networks (colored boxes I, II, III in Fig. 1(c)): a neural oscillator network (I), a phase switching network (PSN, II), and two velocity regulating networks (VRNs, III). Here, we discuss only main functions of the networks (see [4] for a complete description). The neural oscillator network, serving as a central pattern generator (CPG) [7], generates periodic output signals. These signals are provided to all CTr-joints and FTi-joints only indirectly passing through all hidden neurons of the PSN. TC-joints are regulated via the VRNs. Thus, the basic rhythmic leg movement is generated by the neural oscillator network and the steering capability of the walking machine is realized by the PSN and the VRNs.

The neural oscillator network consists of two neurons $H_{1,2}$ with full connectivity. Its synaptic weights and bias terms are selected in accordance with the dynamics of the 2-neuron system [8] staying near the Neimark-Sacker bifurcation where the quasi-periodic attractors occur. The example of periodic output signals having different frequencies resulting from different weights can be seen at [4]. The network has the capability to generate various sinusoidal outputs depending on the weights. Here we use the output signal $v$ of the neural learning circuit (Fig. 1) to modify all weights of the network determined by $w_{11,22} = 0.75v+1.125$, $w_{12} = 1.5v-0.35$, and $w_{21} = -1.5v+0.35$. As a consequence, walking frequency of the machine will be increased by the activation of $v$, resulting in a fast walking speed during escaping from the predator attack.

The PSN is a generic feed-forward network, which reverses the phase of the periodic signals driving the CTr- and FTi-joints (Fig. 1(c), see also [4]). These periodic signals can be switched to lead or lag behind each other by $\pi/2$ in phase in accordance with the given input $I_3$ (Fig. 1(c)). The PSN has been implemented to allow for sideways walking, e.g., for obstacle avoidance (see [4] for more details on parameters and experiments on sensor-driven sideways walking).

The two VRNs are also simple feed-forward networks (see [4]). Each VRN controls the three ipsilateral TC-joints on one side (Fig. 1(c), see also [4]). Because the VRNs behave qualitatively like a multiplication function [4], they have capability to increase or decrease the amplitude of the periodic signals by the magnitude of the inputs $I_{4,5}$. Consequently, the walking speed of the machine will be regulated, i.e., the higher the amplitude of the signal the faster it walks. Therefore, we also apply the neural learning output $v$ to $I_{4,5}$ (see Fig. 1(c) and equations below). Such that the amplitude of the TC-joint signals will be amplified for the fast walking behavior while escaping. Moreover these VRNs can be used to achieve more walking directions, like forward and backward movement (sign inversion of the multiplication) or turning left or right where the directions are driven by other preprocessed infrared and light dependent resistor sensor signals through also $I_{4,5}$ (not shown in the current set of experiments but see [4]).

Figure 1(c) shows the complete network structure together with the synaptic weights of the connections between the controller and the corresponding motor neurons as well as the bias term of each motor neuron. These synaptic weights and all bias terms were manually constructed and adjusted to obtain an optimal gait; i.e., a typical tripod gait [4]. Taken together, this modular neural control can generate more than 10 different walking patterns including various reactive behaviors controlled via the input neurons $I_{1,...,5}$ (Fig. 1(c)). Here, we only focus on the acoustic predator-recognition learning for the safe escape; i.e., the walking machine will perform fast forward walking behavior as soon as it recognizes sound (after learning) or detects the close objects (before/during learning) from behind. Thus, the input neurons $I_{1,...,3}$ are here set to $I_1 = 0.0$, $I_2 = 1.0$, $I_3 = 1.0$, $I_4 = -1.0 + v$, $I_5 = -1.0 + v$ allowing the machine to walk forward only in order to clearly observe the difference between the slow walking speed under normal condition and the fast one during escaping the attack.

## 4 Experiments and Results

In this section, we illustrate the acoustic predator-recognition learning experiments. As shown in Fig. 2, usually before learning AMOS walks fast (≈ 17 cm/s) only if it detects the close objects (here, portable speakers manually moved) from behind through the rear $IR_7$ sensor. During learning, detecting the 300 Hz acoustic signal via the AW sensor and afterwards the objects from behind hints here danger from the predator attack. This situation leads to the correlation between the acoustic signal (predictive signal or "conditioned stimulus" (CS)) and the infrared signal (reflex signal or "unconditioned stimulus" (US)), such that the weight of the plastic synapse $\rho_1$ (Fig. 1(b)) increases. While the weight grows influencing the output $v$ of the learner neuron, AMOS starts to walk fast driven by the CS instead of the US. After learning, it immediately performs fast walking when "hearing" the approaching predator (learned acoustic predator recognition) this way triggering an earlier escape reaction resulting in the safe predator avoidance. The learning process stops when the

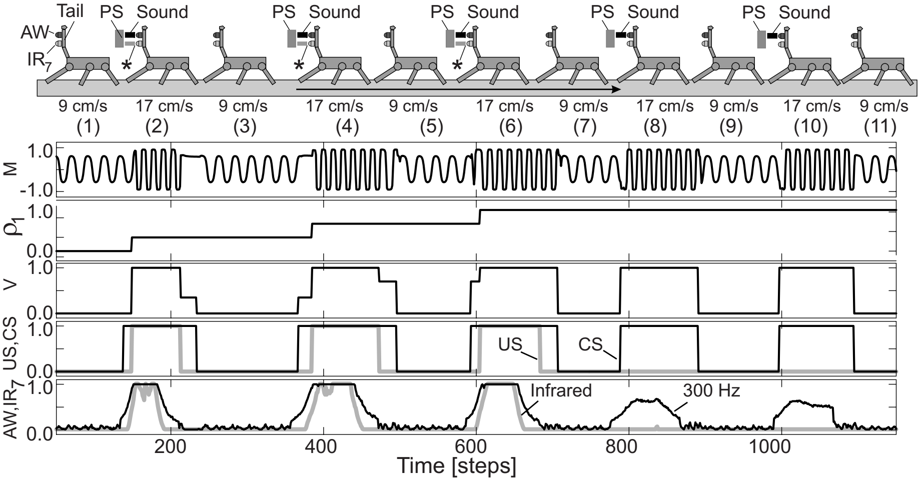


**Fig. 2.** Acoustic predator-recognition learning. The 300 Hz acoustic signal is detected by the auditory-wind detector (AW) sensor while the objects (portable speakers, PS) are detected by the rear infrared ($IR_7$) sensor. In the learning process, AMOS initially walks with its normal speed (1) and then speedily escapes from the too-near objects (2). As soon as it is far enough from the objects it then returns to its normal walking speed (3). Such learning situations repeatedly occur (4-7). Simultaneously, the correlation between the acoustic and infrared signals is being learned. Finally after 3 attacks (1-7) AMOS can immediately react to the acoustic signal as soon as it detects (8-11). Symbols: M = motor signal of a TC-joint, $\rho_1$ = plastic synapse, $v$ = output of a learner neuron, US = unconditioned stimulus (infrared signal), CS = conditioned stimulus (acoustic signal), $*$ = detecting the PS (sound source). Note that one-trial learning could be also achieved by simply increasing a learning rate.

US is no longer triggered which happens when the fast walking behavior is driven by the CS. The video clip of the learning experiments can be seen at http://www.nld.ds.mpg.de/∼poramate/ICONIP09/AMOSLearning.mpg.

## 5 Conclusion

In this study, we simulate acoustic predator-recognition learning on a six-legged walking machine such that the machine can learn the correlation between the predictive acoustic signal ("conditioned stimulus") and the reflex infrared signal ("unconditioned stimulus"). As a consequence, after learning, it performs an earlier escape reaction resulting in the safe predator avoidance as soon as it hears the approaching predator. These learned acoustic predator recognition and escape behavior are achieved under adaptive sensor-driven neural control (Fig. 1) which is modular structure-based design. It consists of three main modules: 1) neural preprocessing unit utilizing recurrent neurodynamics for preprocessing sensory signals, 2) neural learning unit using a correlation based differential Hebbian learning rule for associative learning capabilities, and 3) modular neural control employing a central pattern generator for basic locomotion generation. More demanding tasks will be related to implement neural memory to enhance learning capability through long-term memorization of walking relevant events.

## Acknowledgements

We thank Christoph Kolodziejski for technique advises. This research was supported by the PACO-PLUS project (IST-FP6-IP-027657) as well as by BMBF (Federal Ministry of Education and Research), BCCN (Bernstein Center for Computational Neuroscience)–Göttingen W3.

## References

1. Kindermann, T., Siemers, B.M., Fendt, M.: Innate or Learned Acoustic Recognition of Avian Predators in Rodents? J. Exp. Biol. 212, 506–513 (2009)
2. Pavlov, I.P.: Conditioned reflexes. Oxford University Press, London (1927)
3. Manoonpong, P.: Neural Preprocessing and Control of Reactive Walking Machines: Towards Versatile Artificial Perception-Action Systems. In: Cognitive Technologies. Springer, Heidelberg (2007)
4. Manoonpong, P., Pasemann, F., Wörgötter, F.: Sensor-Driven Neural Control for Omnidirectional Locomotion and Versatile Reactive Behaviors of Walking Machines. Robot. Auton. Syst. 56(3), 265–288 (2008)
5. Manoonpong, P., Pasemann, F., Wörgötter, F.: Neural Preprocessing of Auditory-Wind Sensory Signals and Modular Neural Control for Auditory- and Wind-Evoked Escape Responses of Walking Machines. In: 2008 IEEE International Conference on Robotics and Biomimetics, pp. 786–793. IEEE Press, Los Alamitos (2008)

6. Porr, B., Wörgötter, F.: Strongly Improved Stability and Faster Convergence of Temporal Sequence Learning by Using Input Correlations Only. Neural Comput. 18(6), 1380–1412 (2006)
7. Ijspeert, A.J.: Central Pattern Generators for Locomotion Control in Animals and Robots: A Review. Neural Networks 21, 642–653 (2008)
8. Pasemann, F., Hild, M., Zahedi, K.: SO(2)-Networks as Neural Oscillators. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2686, pp. 144–151. Springer, Heidelberg (2003)

# A Method to Switch Multiple CAN2s for Variable Initial Temperature in Temperature Control of RCA Cleaning Solutions

Shuichi Kurogi, Hiroshi Yuno, and Yohei Koshiyama

Kyushu Institute of technology, Tobata, Kitakyushu, Fukuoka 804-8550, Japan
{kuro@,yuno@kurolab.}cntl.kyutech.ac.jp
http://kurolab.cntl.kyutech.ac.jp/

**Abstract.** The RCA cleaning method is the industry standard way to clean silicon wafers, where the temperature control is important for a stable cleaning performance. However, it is difficult mainly because the RCA solutions expose nonlinear and time-varying exothermic chemical reactions. So far, the MSPC (model switching predictive controller) using the CAN2 (competitive associative net 2) has been developed and the effectiveness has been validated. In this article, we focus on the problem of variable initial temperature which changes the plant dynamics. To solve this problem, we present a method to switch multiple CAN2s, each of which has been trained with the data for different initial temperature. The effectiveness of the present method is evaluated by means of computer simulation.

**Keywords:** switching of competitive associative nets, temperature control of RCA cleaning solutions, variable initial temperature of time-varying plant.

## 1 Introduction

The competitive associative net called CAN2 is a neural net for learning an efficient piecewise linear approximation of nonlinear functions by means of utilizing the competitive and associative schemes[1,2]. The effectiveness of this approach is shown in several areas such as control problems, function approximation, rainfall estimation, time series prediction, and so on [3]-[6]. Here, note that the CAN2 has the following differences from other similar methods; the local linear models [7,8] also utilize the piecewise linear approximation, but they use linear models in piecewise regions obtained via the $K$-nearest neighbors or the SOM (Self-Organizing Map), while the CAN2 utilizes linear models (associative memories) in the piecewise regions obtained via the competitive learning designed for minimizing the mean square error of function approximation. Here, the $K$-nearest neighbors and the SOM are for minimizing the distance measures between input vectors and the centers of the piecewise regions, and do not relate to the mean

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 56–64, 2009.

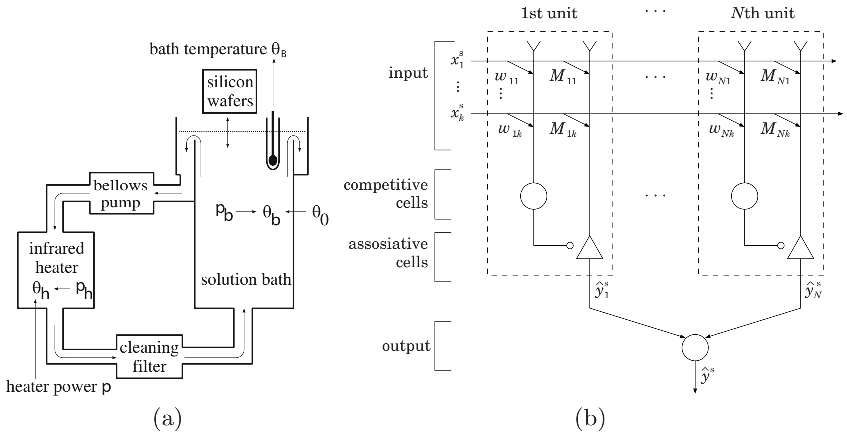


**Fig. 1.** Schematic diagram of (a) the RCA cleaning system and (b) the CAN2

square error of function approximation. Thus, the CAN2 is supposed to show better performance in function approximation and its applications.

On the other hand, the RCA cleaning method introduced and developed at Radio Cooperation of America (RCA) is the industry standard way to clean silicon wafers, and the temperature control is important for a stable cleaning performance. Since the method uses corrosive and hazardous chemical solutions such as SPM (sulfuric acid, $H_2SO_4$, and hydrogen peroxide, $H_2O_2$, mixture) and so on, several pieces of special equipment are arranged for heating solutions as shown in Fig.1(a), where there is a solution bath, a bellows pump, an infrared (IR) heater, and a cleaning filter, which are connected by anti-corrosive recirculation pipes; and the thermal sensor is covered by an anti-corrosive glass tube. Thus, this system involves long and fluctuating time lags and delays, and the mixture of the solutions causes several exothermic chemical reactions which are nonlinear and time-varying. In order to control such nonlinear and time-varying plants involving time lags and delays, we have developed a control method called MSPC (model-switching predictive controller) using the CAN2. Precisely, the CAN2 in the MSPC learns multiple linear models of the plant dynamics from the input and output data of the plant, and then selects an appropriate linear model at each time of the control phase in order for the GPC (generalized predictive controller) to use the selected linear model.

Although we have achieved a good control performance with the CAN2 after learning the training data for a certain initial temperature of the RCA solution so far, we try to cope with the next problem that the initial temperature of the SPM solution varies widely owing mainly to the amount of the remaining solution (water) in the recirculation pipe which is used in the previous cleaning procedure. Here, the perfect removal of the used solution is not so easy for the current RCA plant, and the difficulty of this problem lies in that the dynamics of the solution is different for different initial temperature (see Sect. 3.1 for details).

The remainder of this article is organized as follows; Section 2 gives a brief overview of the MSPC using the CAN2. Section 3 shows the proposed method after examining the variable initial temperature. Section 4 shows numerical experiments and examines the effectiveness.

## 2 MSPC Using First-Order Difference Signals

We briefly show the MSPC using the first-order difference signals[6]. Here, note that the MSPC using the original input and output values of the RCA plant [4] does not work so much in the unit selection process [6] which is crucial for the present method shown in Sect. 3.

### 2.1 Plant Model Using First-Order Difference Signals

In the RCA plant (Fig.1(a)), the input power $p = p(t)$ and the bath temperature $\theta_B = \theta_B(t)$ are sampled with a virtual sampling period $T_v$ as denoted by $u(j) = p(jT_v)$ and $y(j) = \theta_B(jT_v)$ for $j = 1, 2, \cdots, n$. We suppose that they have the relationship or the dynamics given by

$$y(j) = f(\boldsymbol{x}(j)) + d(j) \tag{1}$$

where $f(\cdot)$ denotes a function which is supposed to change slowly (or not to change) in time, and $d(j)$ represents zero-mean noise with the variance $\sigma_d^2$. The input vector $\boldsymbol{x}(j, k_y, k_u) \in X \triangleq \mathbb{R}^{k\times 1}$ is given by $\boldsymbol{x}(j) \triangleq (y(j-1), \cdots, y(j-k_y), u(j-1), \cdots, u(j-k_u))^T$, where $k_y$ and $k_u$ are positive integers, and $k = k_y + k_u$. Then, among the difference signals $\Delta y(j) \triangleq y(j) - y(j-1)$, $\Delta u(j) \triangleq u(j) - u(j-1)$, and $\Delta\boldsymbol{x}(j) \triangleq (\Delta y(j-1), \cdots, \Delta y(j-k_y), \Delta u(j-1), \cdots, \Delta u(j-k_u))$, we have the relationship

$$\Delta y(j) \simeq \boldsymbol{a}^T \Delta\boldsymbol{x}(j), \tag{2}$$

where $\boldsymbol{a} = (a_1, \cdots, a_k)^T \triangleq \partial f(\boldsymbol{x}(j))/\partial\boldsymbol{x}$. If $\boldsymbol{a}$ does not change for a while around the time $j$, we have the following prediction consecutively for $l = 1, 2, \cdots$ as

$$\hat{y}(j+l) = y(j) + \sum_{m=1}^{l} \Delta\widehat{y}(j+m) = y(j) + \sum_{m=1}^{l} \boldsymbol{a}^T \Delta\boldsymbol{x}(j+m), \tag{3}$$

where $\Delta y(j+h)$ in $\Delta\boldsymbol{x}(j+m)$ for $h \geq 1$ is replaced by $\widehat{\Delta y}(j+h)$.

### 2.2 CAN2 for the Difference Plant Model

A CAN2 has $N$ units. The $i$th unit has a weight vector $\boldsymbol{w}_i \triangleq (w_{i1}, \cdots, w_{ik})^T \in \mathbb{R}^{k\times 1}$ and an associative matrix (row vector) $\boldsymbol{M}_i \triangleq (M_{i1}, \cdots, M_{ik}) \in \mathbb{R}^{1\times k}$ for $i \in I = \{1, 2, \cdots, N\}$ (see Fig.1(b)). The CAN2 approximates the output value $y^{\mathrm{s}} = \Delta y(j)$ for the input vector $\boldsymbol{x}^{\mathrm{s}} = \Delta\boldsymbol{x}(j)$ by

$$\widehat{y}^{\mathrm{s}} \triangleq \boldsymbol{M}_c\, \boldsymbol{x}^{\mathrm{s}}, \quad \text{or} \quad \widehat{\Delta y}(j) = \boldsymbol{M}_c \Delta\boldsymbol{x}(j), \tag{4}$$

where the index $c$ is selected by

$$c = \underset{i \in I}{\operatorname{argmin}} \, \|\boldsymbol{x}^{\mathrm{s}} - \boldsymbol{w}_i\|^2. \tag{5}$$

Here, the variables $\boldsymbol{x}^{\mathrm{s}}$ and $y^{\mathrm{s}}$ represent the input and the output of the CAN2, respectively, and their superscript "s" indicates the variable related to the CAN2 for distinguishing $\boldsymbol{x}(j)$ and $y(j)$ of the RCA system.

### 2.3 MSPC for the Difference Plant Model

The MSPC utilizes the CAN2 as well as the GPC (Generalized Predictive Control) for obtaining the predictive input $\widehat{u}(j)$ which minimizes the control performance index:

$$J = \sum_{l=N_1}^{N_2} \widehat{e}^2(j+l) + \lambda_u \sum_{l=1}^{N_u} \widehat{\Delta u}^2(j+l-1), \tag{6}$$

where $\widehat{e}(j+l) = y_d(j+l) - \widehat{y}(j+l)$ indicates the error between the desired output $y_d(j+l)$ and the predictive output $\widehat{y}(j+l)$. The parameters $N_1$, $N_2$, $N_u$ and $\lambda_u$ are constants to be designed for the control performance. We obtain $\widehat{\Delta u}(j+l)$ for $l = 0, 1, \cdots$ as follows; after learning the pairs $(\boldsymbol{x}^{\mathrm{s}}, y^{\mathrm{s}}) = (\Delta\boldsymbol{x}(j), \Delta y(j))$ for the training dataset $D^n$ obtained at the previous control phase (see the next section), the CAN2 at a discrete time $j$ during the current control phase can predict $\Delta y(j+l)$ for $l = 1, 2, \cdots$ consecutively by Eq.(4), and $\hat{y}(j+l)$ by Eq.(3) whose $\boldsymbol{a}^T$ is replaced by $\boldsymbol{M}_c$. From the GPC method with this equation, the above performance index is modified to

$$J = \|\boldsymbol{y}_d - G\widehat{\Delta\boldsymbol{u}} - \boldsymbol{p}\|^2 + \lambda_u \|\widehat{\Delta\boldsymbol{u}}\|^2 \tag{7}$$

where $\boldsymbol{y}_d = (y_d(j+N_1), \cdots, y_d(j+N_2))^T$, $\widehat{\Delta\boldsymbol{u}} = (\widehat{\Delta u}(j), \cdots, \widehat{\Delta u}(j+N_u-1))^T$, and $\boldsymbol{p} = (p(j+N_1), \cdots, p(j+N_2))^T$. Here, $p(j+l)$ is the natural response $\widehat{y}(j+l)$ of the system Eq.(3) for the null incremental input $\widehat{\Delta u}(j+l) = 0$ for $l \geq 0$. The $i$th column and the $j$th row of the matrix $\boldsymbol{G}$ is given by $G_{ij} = g_{i-j+N_1}$, where $g_l$ for $l = \cdots, -2, -1, 0, 1, 2, \cdots$ is the unit step response $\widehat{y}(j+l)$ of Eq.(3) for $\widehat{y}(j-l) = \widehat{u}(j-l) = 0$ $(l < 0)$ and $\widehat{u}(l) = 1 (l \geq 0)$. Then, $J$ is minimized by

$$\widehat{\Delta\boldsymbol{u}} = (\boldsymbol{G}^T\boldsymbol{G} + \lambda_u \boldsymbol{I})^{-1}\boldsymbol{G}^T(\boldsymbol{y}_d - \boldsymbol{p}), \tag{8}$$

and we have $\widehat{u}(j) = u(j-1) + \widehat{\Delta u}(j)$.

### 2.4 Iteration of Control and Learning

Before applying the MSPC to the real RCA plant, we train the CAN2 with the data obtained from a simulation result using the differential equation model of the RCA plant [3,4]. Namely, with certain initial values of $\boldsymbol{w}_i$ and $\boldsymbol{M}_i$ $(i \in I)$, we execute the following two phases repeatedly;

(i) **Control phase:** Control the differential model RCA plant, and obtain the dataset $D^n = \{(\boldsymbol{x}(j), y(j) | j = 1, 2, \cdots, n)\}$ involving the input $\boldsymbol{x}(j)$ and the output $y(j)$ of the RCA system.

(ii) **Learning phase:** Apply the learning method to the CAN2 with $D^n$, where $\boldsymbol{w}_i$ and $\boldsymbol{M}_i$ are updated to minimize the approximation error on $D^n$.

# 3 Method to Cope with Variable Initial Temperature

## 3.1 Property of Variable Initial Temperature

As described in Sect. 1, the initial temperature $\theta_0$ of the SPM varies owing mainly to the remaining solution used for the previous cleaning process, where $\theta_0$ rises by means of the exothermic reaction between the water in the remaining used solution and the sulfuric acid in the new SPM. The exothermic chemical reaction of the SPM through the temperature control depends on $\theta_0$ and the time after mixing the constituent solutions of the SPM. Here, note that $\theta_0$ usually varies from 80°C to 120°C, and a specific objective of the temperature control is to rise the bath temperature to the set point $\theta_B = 135$°C with allowable error ±2°C and to reduce the settling time $T_{\rm S}$ as short as possible. Since the chemical reaction is more active for higher $\theta_0$, the control of $\theta_{\rm O}$ is more difficult for a higher $\theta_0$ and easier for lower $\theta_0$.

## 3.2 Switching of CAN2s for Variable Initial Temperature

In order to cope with the above problem from the viewpoint of the MSPC, it may be natural to prepare multiple CAN2s for several initial temperatures, and select an appropriate CAN2 at each time in control phase. Precisely, we can do the following steps with multiple CAN2s, or CAN2$^{\theta_0^s}$ ($s \in S = \{1, 2, \cdots, |S|\}$) each of which has been trained with the dataset for the initial temperature $\theta_0^s$.

**step 1:** At the time $j$ in the control phase, select the unit for each CAN2$^{\theta_0^s}$ by Eq.(5), or

$$c_j^s = \mathop{\rm argmin}_{i \in I} \|\Delta\boldsymbol{x}(j) - \boldsymbol{w}_i^s\|^2 \tag{9}$$

where $\boldsymbol{w}_i^s$ ($i \in I$) are the weight vectors of the $s$th CAN2, or CAN2$^{\theta_0^s}$.

**step 2:** Select the $s^*$th CAN2 which has the minimum sum of the square prediction error for the time $j - l + 1$ ($l = 1, \cdots, N_e$), or

$$s_j^* = \mathop{\rm argmin}_{s \in S} \sum_{l=1}^{N_e} \|\Delta y(j - l + 1) - \boldsymbol{M}_{c_j^s}^s \Delta\boldsymbol{x}(j - l + 1)\|^2, \tag{10}$$

where $\boldsymbol{M}_{c_j^s}^s$ is the $c_j^s$th associative matrix of the $s$th CAN2.

Note that **step 2** evaluates $\boldsymbol{M}_{c_j^s}^s$ selected by **step 1** for not only time $j$ but also $j - 1, j - 2, \cdots$, and $\boldsymbol{M}_{c_j^s}^s$ for the selected $s = s_j^*$th CAN2 is used for predicting the output for the time $j + 1, j + 2, \cdots$ by means of the GPC.

## 4 Numerical Experiments

We have examined the present method with the differential model of the RCA plant [3,4] whose parameter values are set for the SPM. The control procedure is supposed to last 8000s and the sampling period is set $T = 0.25$s, so that the number of the training data is about $n = 32,000 = 8,000/0.25$. The dimension of the input vector $\boldsymbol{x}^{\mathrm{s}} = \varDelta\boldsymbol{x}(j)$ is set $k = 6$ ($k_y = 3$ and $k_u = 3$), and the parameter values in Eq.(7) are set $\lambda_u = 0.22$, $N_1 = 1$, $N_2 = 11$ and $N_u = 1$. We prepare two CAN2s, i.e., CAN2$^{80}$ and CAN2$^{120}$ trained for the initial temperatures 80°C and 120°C, respectively, where we execute 30 iterations of the control and learning phases. The parameter $N_e$ (see Sect. 3.2) of the present method is set $N_e = 1$ (we omit the result for $N_e > 1$ because of the limitation of page numbers). We denote CAN2$^{120+80}$ for the proposed method to switch CAN2$^{120}$ and CAN2$^{80}$.

### 4.1 Overshoot and Settling Time for Different Initial Temperatures

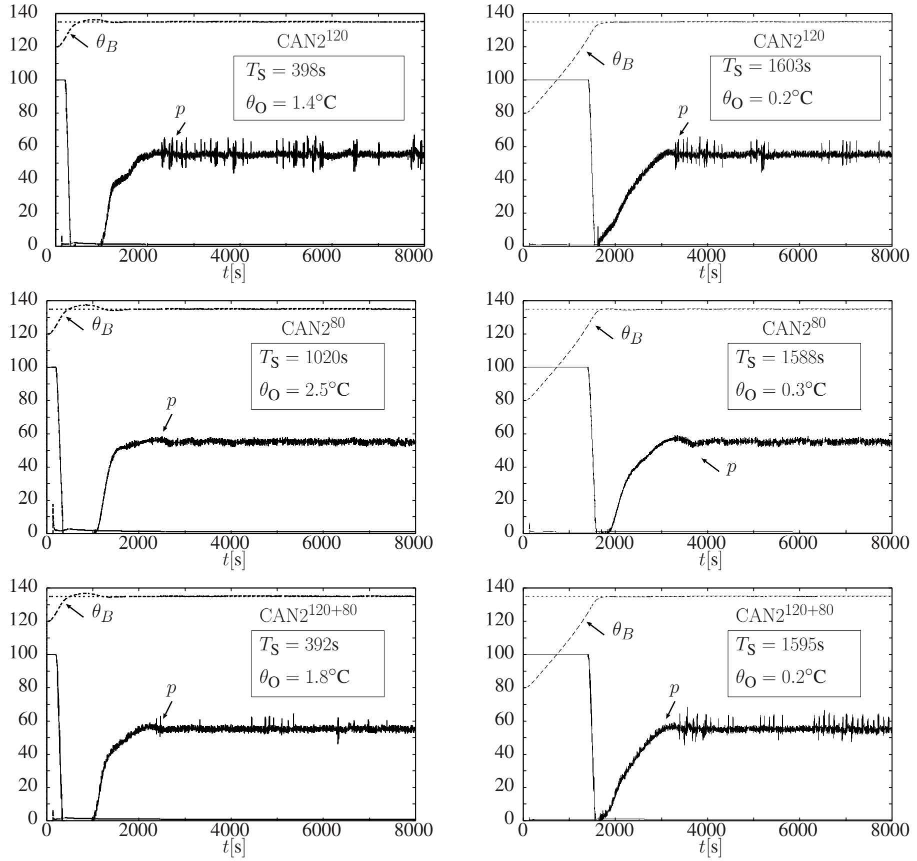
The time course of the input power $p$[%], the output bath temperature $\theta_B$[°C] for the initial temperature $\theta_0 = 120$°C and 80°C controlled by the MSPC using CAN2$^{120}$, CAN2$^{80}$ and CAN2$^{120+80}$, respectively, are shown in Fig.2. First of all, we can see that $\theta_B$ increases even if $p$ is small from $t = 500$ to 1000 for $\theta_0 = 120$ (left) and around $t = 1800$ for $\theta_0 = 80$ (right), which indicates that exothermic chemical reaction is occurring, while $p$ should be kept about 60% (of the maximum power) at $t > 3000$ where the exothermic reaction is disappeared.

The summary of $\theta_O$ and $T_{\mathrm{S}}$ of the above cases as well as those for $\theta_0 = 90$, 100, 110 is shown in Table 1. From the table on the left hand side of Table 1, we can see that CAN2$^{80}$ could not have achieved $\theta_O \leq$2°C for $\theta_0 = 120$°C. However, CAN2$^{120+80}$ as well as CAN2$^{120}$ have achieved $\theta_O \leq$2°C for both $\theta_0 = 120$°C and 80°C. This result is reasonable because the chemical exothermic reaction for $\theta_0 = 120$°C is bigger than the lower $\theta_0$ as shown in Sect. 3.1, so that the method achieving $\theta_O \leq$2°C for $\theta_0 = 120$°C is supposed to be able to achieve $\theta_O \leq$2°C for lower $\theta_0$. From two tables in Table 1, we can see that CAN2$^{120+80}$ has achieved $\theta_O \leq$2°C for all $\theta_0$, and whose $T_{\mathrm{S}}$ is almost the same as $T_{\mathrm{S}}$ by CAN2$^{\theta_0}$ which is trained with the data for $\theta_0$, respectively.

**Table 1.** The overshoot $\theta_{\mathrm{O}}$[°C] and the settling time $T_{\mathrm{S}}$[s] for the initial temperature $\theta_0$[°C] by means of CAN2$^{120}$, CAN2$^{80}$, CAN2$^{120+80}$ and CAN2$^{\theta_0}$ for $\theta_0 = 110, 100, 90$

| | CAN2$^{120}$ | | CAN2$^{80}$ | | CAN2$^{120+80}$ | |
|---|---|---|---|---|---|---|
| $\theta_0$ | $\theta_{\mathrm{O}}$ | $T_{\mathrm{S}}$ | $\theta_{\mathrm{O}}$ | $T_{\mathrm{S}}$ | $\theta_{\mathrm{O}}$ | $T_{\mathrm{S}}$ |
| 120 | 1.4 | 398 | 2.5 | 1020 | 1.8 | 392 |
| 80 | 0.2 | 1603 | 0.3 | 1588 | 0.2 | 1595 |

| | CAN2$^{\theta_0}$ | | CAN2$^{120+80}$ | |
|---|---|---|---|---|
| $\theta_0$ | $\theta_{\mathrm{O}}$ | $T_{\mathrm{S}}$ | $\theta_{\mathrm{O}}$ | $T_{\mathrm{S}}$ |
| 110 | 1.0 | 665 | 0.6 | 659 |
| 100 | 0.7 | 956 | 0.2 | 965 |
| 90 | 0.1 | 1276 | 0.2 | 1282 |



(a) $\theta_0 = 120$°C

(b) $\theta_0 = 80$°C

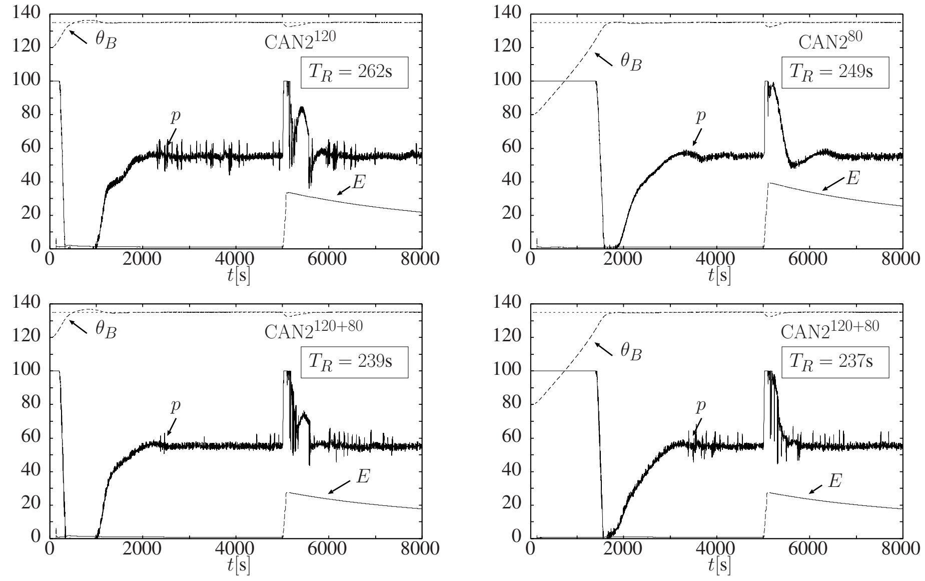
**Fig. 2.** Examples of the control result

### 4.2 Recovery from the Disturbance

After the bath temperature reaches the set point, the procedure to clean silicon wafers is executed, which is modeled by a temperature disturbance accompanied with the modification of the specific heat coefficient. The temperature should be recovered quickly. Two examples are shown in Fig.3, and they indicate that the present method using CAN2$^{120+80}$ has achieved smaller recovery time than CAN2$^{120}$ and CAN2$^{80}$, respectively, for each case.

## 5 Conclusion

We have focused on the problem of variable initial temperature of the RCA cleaning plant, and presented a method to switch multiple CAN2s, each of which has



(a) $\theta_0 = 120°$C

(b) $\theta_0 = 80°$C

**Fig. 3.** Recovery from the disturbance occurred at $t = 5000$s. $T_R$: the recovery time.

been trained with the data for different initial temperatures. The effectiveness of the present method is examined by computer simulation. The result shows that the present method works well for every initial temperature from the point of view of the overshoot, the settling time and the disturbance recovery. We would like to examine the robustness and the generality of the method in various cases in our future research study.

**Acknowledgments.** This work was partially supported by the Grant-in Aid for Scientific Research (C) 21500217 of the Japanese Ministry of Education, Science, Sports and Culture.

## References

1. Kohonen, T.: Associative Memory. Springer, Heidelberg (1977)
2. Ahalt, A.C., Krishnamurthy, A.K., Chen, P., Melton, D.E.: Competitive learning algorithms for vector quantization. Neural Networks 3, 277–290 (1990)
3. Kurogi, S., Nishida, T., Sakamoto, T., Itoh, K., Mimata, M.: A simplified competitive associative net and a model-switching predictive controller for temperature control of chemical solutions. In: Proc. ICONIP 2000, pp. 791–796 (2000)
4. Kurogi, S., Nishida, T., Nobutomo, H., Sakamoto, T., Mimata, M., Itoh, K.: A thermal model of the RCA cleaning system and adaptive predictive temperature control of cleaning solutions. Trans. SICE (in Japanese) 37(8), 754–762 (2001)

5. http://predict.kyb.tuebingen.mpg.de/pages/home.php
6. Kurogi, S., Koshiyama, Y., Kuwahara, D., Nishida, T.: Model switching predictive control using bagging CAN2 and first-difference signals for temperature control of RCA cleaning solutions. In: Proc. WCCI 2008, pp. 2322–2387 (2008)
7. Farmer, J.D., Sidorowich, J.J.: Predicting chaotic time series. Phys. Rev. Lett. 59, 845–848 (1987)
8. Ritter, H., Martinetz, T., Schulten, K.: Neural computation and self-organizing maps. Addison and Wesley, London (1992)

# Vision-Motor Abstraction toward Robot Cognition

Fady Alnajjar, Abdul Rahman Hafiz, Indra Bin Mohd Zin, and Kazuyuki Murase

Graduate School of Engineering, University of Fukui, Japan
{fady,abdul,indra,murase}@synapse.his.fukui-u.ac.jp

**Abstract.** Based on indications from neuroscience and psychology, both perception and action can be internally simulated in organisms by activating sensory and/or motor areas in the brain without actual sensory input and/or without any resulting behavior. This phenomenon is usually used by the organisms to cope with missing external inputs. Applying such a phenomenon in a real robot recently has taken the attention of many researchers. Although some work has been reported on this issue, none of it has so far considered the potential of the robot's vision at the sensorimotor abstraction level, where extracting data from the environment to build the internal representation takes place. In this study, a novel vision-motor abstraction is presented into a physically robot through a memory-based learning algorithm. Experimental results indicate that our robot with its vision could develop a simple anticipation mechanism in its memory from the interacting with the environment. This mechanism could guide the robot behavior in the absence of external inputs.

**Keywords:** Vision-motor abstraction, memory based learning, cognition.

## 1 Introduction

Real world applications are usually subject to change and very difficult to be predicted. Any sudden changes in the environment can possibly cause temporary lose in communication with the external world. Some organisms, those that have the ability of cognition, can cope with such situations by replacing the external missing sensory data with their own internal representation.

In recent decades, a branch of science called cognitive neuroscience has been established to introduce such a phenomenon to the robot [1]. It is concerned with endowing robots with human-like cognitive capabilities to enable them to accomplish complex tasks in complex environments. In [2], the authors have argued that all living creatures are cognitive to some degree, and therefore, it could be achieved to some extent on the robots [8]. We believe that the level of or how much the robot could be conscious of the surrounding environment depends on how much the robot knows about this environment. One of the possible approaches to measure robot's consciousness is by examining its ability to cope with missing external sensory data during performance of a specific task, i.e., performs blindfolded navigation using only its internal representation.

In recent years therefore, building a complete blindfolded navigation system in a robot has been a challenging task [3,4,6]. For instance, some initial experiments were

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 65–74, 2009.

presented in [3] that aim to contribute toward building a robot that navigates completely blindfolded in a simple environment using a two-level network architecture; (a) low-level abstraction from sensorimotor values to a limited number of simple concepts, following the work done by Linker and Niklasson [4], and (b) higher-level prediction/representation of the agent's interaction with the environment, inspired by the work done by Nolfi and Tani [5]. These efforts have succeeded in allowing the robot to anticipate long chains of future situations. However, they have failed to support a completely blindfolded navigation [6], in which the robot repeatedly uses only its internal representation values for its navigation. The failure partly seems to be due to the short range of the robot's proximity sensors that they used, which limits the data that could be abstracted from the environment. We argue here that improving the robot's sensorimotor abstraction level from the limited short range sensor to a wide vision view could possibly overcome this problem.

To support our argument, we have done psychological experiments, similar to the one introduced by Lee and Thompson [7]. In a series of two experiments, we demonstrated the accuracy with which humans can guide their behavior based only on their internally generated sensory experiences. Two subjects were asked to do a task under different conditions. The first subject (X) was asked to 'look' around in a room and locate a specific target (Fig.1A). He was then blindfolded and asked to locate the target again. The subject performed the task accurately with closed eyes, in the same manner as when he was free to 'look' (Fig.1B). However, he could not predict the exact time needed to turn to the target and this caused a few hits with the obstacle (the empty circles in Fig.1B). The second subject (Y) was not allowed to explore the room with his eyes (no vision input). Instead, he was blindfolded and walked around the room touching things around him until he found the target (Fig.1C). He was then asked to seek the target again blindfolded from the initial position. Though successful in reaching the target, he took more time than that needed by subject (X). In addition, the number of times that he hit the wall or touched it to correct or locate his direction was greater (the empty circles in Fig.1D).

From the above experiment, we can conclude that subject (X) had collected a sufficient data during his first 'eyes open' navigation. This data could be various dimensions in the room which the subject related to times and distances that helped him to build internally his own image. In contrast, the data that subject (Y) had collected was limited to the objects that his hand touched during his first blindfolded navigation and their relation to his moving steps. This data, however, was not good enough to accurately perform the task.

In the above experiment, subject Y could be a demonstration of the results of the most recently reported works (e.g., [3]), since they used the short-range sensors for building the sensorimotor abstraction level. Our target in this study is to demonstrate the subject X behavior. Here we explore the inner world of a real robot that has a chance to explore the surrounding environment with its camera before it was told to navigate blindfolded in it.

The rest of the paper is structured as follows. Section 2 briefly presents the background. Section 3 describes the robot and the environment. Section 4 explains the architecture of the proposed algorithm and the experiments. The final section presents a brief discussion of the work and gives some future work suggestions.

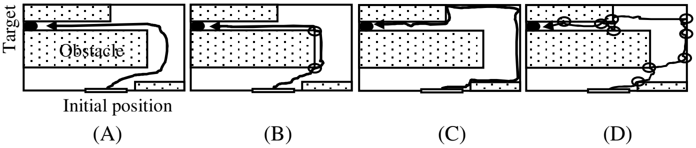


**Fig. 1.** The track of subject X\Y: (A)\(C) the first move, (B)\(D) the second move

## 2 Robot Cognition

A number of researchers have tried to investigate the robot's inner world [8]. Hesslow's paper [8], for instance, described a development of three simulation hypotheses in order to explain the robot's inner world. This was also discussed by [3]: The first is covert behavior, which is the ability to generate internally neural motor responses that are not actually externally executed. The second is sensor imagery, which is the ability to internally activate the sensory areas in the brain, so as to produce the simulated experience without actual inputs. The third is anticipation, which is the ability to predict the sensory consequences of the motor response [3,8].

Based on the above hypotheses, the internal sequences of the robot behaviors could be illustrated by Fig.2. In Fig.2A, a situation S1 elicits internal activity s1, which in turn leads to a motor response preparation r1 and thereafter results in the overt behavior R1, which causes a new situation S2. In Fig.2B, because of the robot's past experiences, the response preparation r1 could directly elicit the internal activity s2. In Fig.2C, if the robot trains the network to some degree, then it should be possible to simulate long sequences of motor responses and sensory consequences.

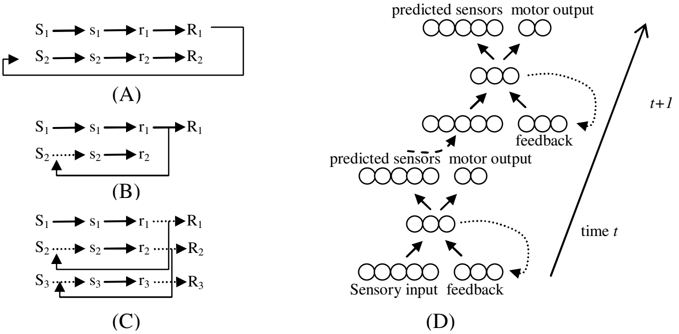


**Fig. 2.** (A)-(C) The basic principle of Hesslow's hypothesis (Adopted from [8]). (D) The basic approach of simulation of perception in robots used by [3].

Several modelers have used the above approach directly into the robot [2,3]. These modelers have mapped the sensory input to motor output, and also predicted the next time step's sensory input based on the network's experience. Most of these works, however, have considered only the robot's short-range sensors (e.g., IR sensor) in their experiments, which cannot provide enough data about the environment for the robot to build its internal representation.

From the psychological experiments in the above section, we agree with [3] that the weak spot in simulation theories is concerning the matter of the abstraction level at which internal representation is relied on. Therefore, improving the ability of this level could result in better internal representation in the robot, and in consequence, better cognition ability.

## 3 Robot and Environment

All the experiments in here were conducted in a physical mobile robot “Hemisson”. (www.k-team.com) (Fig.3A). The robot is able to avoid obstacles, detect ambient light intensity and follow a line on the floor. Hemisson is equipped with IR sensors and a wireless camera to transmit video images to a receiver that is connected to a PC for simple image processing.

In this study, the robot’s camera view has been divided into 4 parts, as illustrated in Fig.3C. Each part covers a number of pixels that represent the distance to the obstacles. The combination of these parts was used to indentify the current concept of the robot’s view (CC). The average of $F_L$ and $F_R$ were used to calculate the real distance (D) cm to the frontal obstacles. We have applied the idea of the flood fill algorithm [9] to filter the robot’s view and to easily clarify the boundaries between the floor and the obstacles. We arranged an ideal environment for the robot to avoid a large amount of image processing. A simple neural network was trained by Back-Propagation (BP) to convert the number of pixels in each part into a real distance.

The environment we used was similar to the one used by [3], consisting of two different-sized rooms connected by a short corridor (Fig.3B).

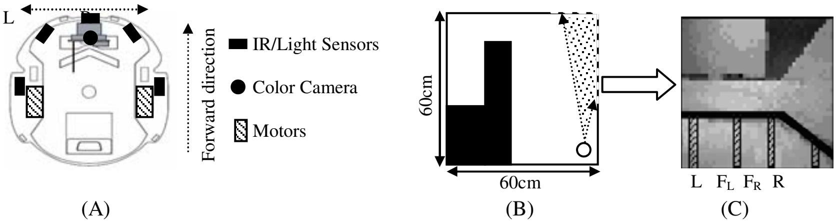


**Fig. 3.** (A) Schematic drawing of Hemisson. (B) Robot environment. The empty circle illustrates the robot. The dots area illustrates the range of robot’s view. (C) Snapshot of the Robot’s view taken from the position shown in B. Black thick line illustrates the lower edge of the obstacle. L, $F_L$, $F_R$ and R, illustrate the left, left-front, right-front and right pixels range reading, respectively.

## 4 Experiments

### 4.1 The Proposed Architecture

The general network architecture in this study was inspired by the architecture in [3]. In their work, they used two-level neural network architecture. The lower level consisted of an unsupervised vector quantizer that categorized the current IR-sensor and motor values into a more abstract level; they called "concepts", such as "corner" or "corridor". The higher level consisted of a recurrent neural network that trained by either BP or by GA to try to predict the sequence of the lower-level concepts and their respective duration [3].

Our architecture, in contrast, differs in two main aspects. First, instead of using only the IR sensors as an input for the abstract level, we added the robot's visual sensors to the system. Second, the robot has two separated networks. The first, to control the robot's navigation (Fig.4A), while the second, to represent the robot's memory (Fig.4B). The second network is used to abstract data from both the first network (motors' speed) and the environment. It also learns the relationships between these data to build the robot's internal representation, and to predict both the upcoming concept (PNC) and the time needed to go through each concept (PT).

We tried to train the second network with both GA and BP, as was suggested by [3,6]. However, the error ratio in both algorithms was very high, even when we trained the robot for a very long time. The reasons for these failures could be that the number of concepts generated by the robot's camera has a sequence which is too complex for such algorithms to learn; especially we are dealing with a physical robot that makes learning through these types of evolutionary algorithms quite impossible.

We, therefore, shifted the learning process in the second network so to be based on the contents of the robot's memory. That is while the robot is navigating in the environment using the first network; the second network tries to memorize the environment and predicts them from its experience the next view or action, and corrects later its data by the actual fact when it faces it. This algorithm turned out to be reasonably successful.

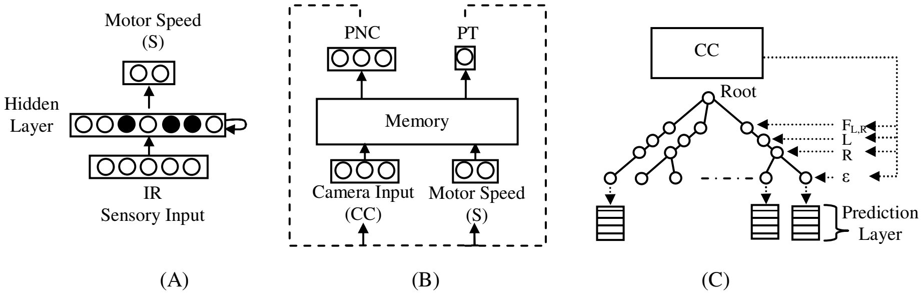


**Fig. 4.** (A) Architecture of the SNN used in the first network. White\black circles represent excitatory\inhibitory neurons which have positive\negative connection. The neurons in the hidden layer are fully connected to each other (Adopted from [10]). (B) Architecture of the second network. *PNC* and *PT* are the output of the robot's memory at each time step. The dashed line illustrates the feedback connection done in experiment 3. (C) Tree-memory structure used in this study (Adopted from [11]). Each concept has its own prediction layer. Prediction layer contains *PNC, PT* and the end action of each concept, e.g., turn right or turn left.

A tree-type memory structure has been introduced to the second network, similar to the one introduced in [11] (Fig.4C). This memory has a dynamic structure and simple storing and retrieving mechanism. It was also supported by forgetting and clustering mechanisms to control its general size and to provide a maximum memorizing ability. The memory was divided into five levels. The first three levels were used to store the robot camera inputs to identify each concept. The fourth level, (ε), was used to count the number of concepts in each environment to identify the environment. The last level represents the prediction layer.

During the navigation, the robot built its memory and used it to predict its future action. The flowchart in Fig.5 shows the working mechanism of the memory. According to the chart, when the robot returns to its straight state after performing the turning action, two phases are operated sequentially: the learning phase (the thick lines in the chart), where the robot learns and updates its memory with the currently available facts, and the predicting phase (the thin lines in the chart), where the robot explores the environment and gradually builds its experiences.

## 4.2 Experimental Results

### 4.2.1 Wall-Following Behavior

In order to control the robot's initial behavior, the robot's first network was programmed to perform wall following behavior using a pre-trained self-organizing spiking neural network [10] Fig.4A.

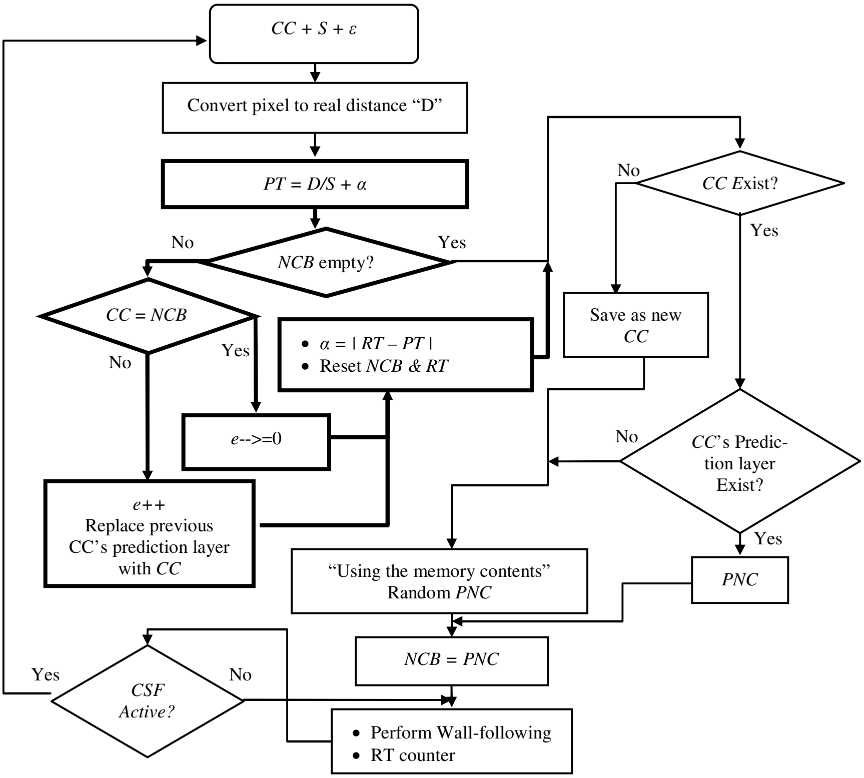


**Fig. 5.** The working mechanism of the robot's memory. Where NCB: next concept buffer. RT: real time. *e:* error prediction rate. CSF: changing-state flag.

As previously stated, the robot used this network exclusively for performing the navigation task; no data abstraction from the environment was processed in this level. The activation of CSF at every new state excited the second network to do its task. To simplify the second network's task, which partly depended on the motor output from the first network, we limited the robot's forward speed to a value equal to the average of the robot's forward speed in 10 success rounds in the environment (S=1.25cm/sec).

### 4.2.2 Data Abstraction and Prediction

The main objective of this stage is to examine the validity of the second network to build an internal representation, so that, the robot can keep tracking its own relative position in the environment and to anticipate the upcoming event.

Similar work to [3] has been carried out in this stage with some variations. The sensorimotor abstract level in the second network was supported by the robot's vision and a tree-memory structure. In the experiment, we left the robot, using the first network, to perform the wall-following task for 5 rounds, simultaneously with the existence of the second network whenever CSF was activated. At the beginning of each concept, the network trained both PNC and PT.

Figure 6A&B show the number of concepts that the robot could identify from the environment using its camera. From the figure, we can see that our method abstracted 7 different concepts from the environment, while in [3] only 5 concepts were found by using the short-range IR sensors. Notice that the number of concepts indicates to what degree the robot is sensitive about the environment, and as a consequence, it would results in better anticipation.

Table 1 shows the evolvement of the learning and predicting phases in the robot's memory during the 3 complete rounds in the environment. From the table, at the first round, the robot is not able to predict neither the PNC nor PT correctly. The robot set these values randomly since it does not have experience about them. In the learning phase, however, it updates both of these values in each concept by learning online from the value of RNC and RT, respectively. It is worthwhile to mention that the robot built a suitable knowledge about the surrounding environment within the first two rounds. Within the 3rd round the robot was able to predict all the PNC correctly, i.e., PNC = RNC and $e$ decreases to 0. Although, the robot was unable to predict PT 100% correctly, i.e., PT≠RT, however, its value comes very close to RT and $\alpha$ turned out to be very close to 0. We have also introduced different environment to the robot; the robot could easily adapt to the new environment and in short time could update its memory to predict the sequence of the new environment (experiment results were not shown due to the limitation).

### 4.2.3 Blindfolded Navigation

The objective of this stage is to examine the ability of the robot to replace all of its external sensory input with its own internal representation, i.e., repeatedly using the sequences of its own prediction for a certain number of times without external sensory input, see the dashed lines in Fig.4B. In other words, have the robot navigates itself blindfolded in the environment.

In this stage, after the robot trained in the original environment for 5 rounds, we removed the surrounding environment completely, eliminated the external sensory inputs, and let the robot move in a wide space using only the last found sequence of the concepts in its memory.

Figure 6C shows the robot's best behavior. Interestingly, the robot built experiences about the environment in its memory sensitive enough so that it could navigate in the environment without any interaction with the external world. All the concepts were memorized correctly, and the robot moved according to the environment's layout. Unfortunately, the robot has slightly shifted its movement in each round, and this was probably due to the error in predicting the time of each concept.

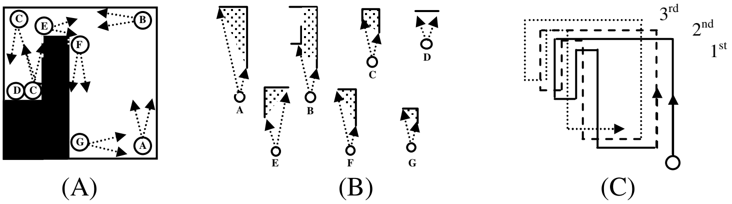


**Fig. 6.** (A) Number of concepts that robot's camera could identify in the original environment. (B) The layout of the robot's view in each concept. (C) Robot's behavior during 3 rounds simulating only its internal representation.

**Table 1.** Second Network Evolvements for 3 Rounds

(Each concept assigned by CC's value that automatically generated in sequential manner). *RNC* = the *CC* of the next step.

| | Predicting phase | | | | | | | | Learning phase | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | *CC* | | *S* | | | | | | | | |
| Round | D(*cm*) | ***CC*** | *Value* | ε | *cm/sec* | *D/S* | *PT* | ***PNC*** | *RT* | *α* | *RNC* | *e* | ***e+α*** |
| 1 | 40.04 | **A** | 1 | 1 | 1.25 | 32.032 | 32.032 | **A** | 39.5 | 7.5 | B | 1 | **8.5** |
| 1 | 40.04 | **B** | 2 | 3 | 1.25 | 32.032 | 32.032 | **A** | 39.1 | 7.1 | C | 2 | **9.1** |
| 1 | 24.88 | **C** | 3 | 6 | 1.25 | 19.904 | 19.904 | **B** | 21.0 | 1.1 | D | 3 | **4.1** |
| 1 | 5.01 | **D** | 4 | 10 | 1.25 | 4.008 | 4.008 | **D** | 5.1 | 1.1 | C | 4 | **5.1** |
| 1 | 24.55 | **C** | 3 | 7 | 1.25 | 19.64 | 19.64 | **D** | 18.5 | -1.1 | E | 5 | **3.9** |
| 1 | 32.65 | **E** | 5 | 12 | 1.25 | 0 | 0 | **B** | 4.5 | 4.5 | F | 6 | **10.5** |
| 1 | 36.8 | **F** | 6 | 18 | 1.25 | 29.44 | 29.44 | **C** | 33.8 | 4.4 | G | 7 | **11.4** |
| 1 | 27.7 | **G** | 7 | 25 | 1.25 | 22.16 | 22.16 | **B** | 21.3 | -0.9 | A | 8 | **7.1** |
| 2 | 39.4 | **A** | 1 | 24 | 1.25 | 31.52 | 39.0 | **B** | 39.1 | 0.1 | B | 7 | **7.1** |
| 2 | 40.04 | **B** | 2 | 22 | 1.25 | 32.032 | 39.1 | **C** | 39.4 | 0.3 | C | 6 | **6.3** |
| 2 | 23.84 | **C** | 3 | 25 | 1.25 | 19.072 | 20.2 | **E** | 20.5 | 0.3 | D | 7 | **7.3** |
| 2 | 5.01 | **D** | 4 | 21 | 1.25 | 4.008 | 5.1 | **C** | 5.1 | 0.0 | C | 6 | **6.0** |
| 2 | 24.88 | **C** | 3 | 18 | 1.25 | 19.904 | 18.8 | **D** | 19.5 | 0.7 | E | 7 | **7.7** |
| 2 | 32.17 | **E** | 5 | 13 | 1.25 | 0 | 4.5 | **F** | 5.0 | 0.5 | F | 6 | **6.5** |
| 2 | 36.06 | **F** | 6 | 7 | 1.25 | 28.848 | 33.2 | **G** | 33.0 | -0.2 | G | 5 | **4.8** |
| 2 | 27 | **G** | 7 | 0 | 1.25 | 21.6 | 20.7 | **A** | 20.4 | -0.4 | A | 4 | **4.4** |
| 3 | 39.4 | **A** | 1 | 1 | 1.25 | 31.52 | 39.1 | **B** | 40.2 | 1.1 | B | 3 | **4.1** |
| 3 | 39.09 | **B** | 2 | 3 | 1.25 | 31.272 | 38.6 | **C** | 38.9 | 0.3 | C | 2 | **2.3** |
| 3 | 24.36 | **C** | 3 | 6 | 1.25 | 19.488 | 20.9 | **D** | 20.5 | -0.4 | D | 1 | **1.4** |
| 3 | 5.01 | **D** | 4 | 10 | 1.25 | 4.008 | 5.1 | **C** | 5.0 | -0.1 | C | 0 | **0.1** |
| 3 | 24.88 | **C** | 3 | 7 | 1.25 | 19.904 | 19.5 | **E** | 18.8 | -0.7 | E | 0 | **0.7** |
| 3 | 36.06 | **E** | 5 | 12 | 1.25 | 0 | 5.0 | **F** | 5.0 | 0.0 | F | 0 | **0.0** |
| 3 | 36.8 | **F** | 6 | 18 | 1.25 | 29.44 | 33.6 | **G** | 33.5 | -0.1 | G | 0 | **0.1** |
| 3 | 27 | **G** | 7 | 25 | 1.25 | 21.6 | 20.4 | **A** | 21.0 | 0.6 | A | 0 | **0.6** |

## 5 Conclusion

We have presented some initial experiments with the aim to contribute toward the development of robot models in sensorimotor abstraction, simulation and anticipation. In particular, and unlike the most previous related work, we have here presented (a) a robot equipped with a video camera to extract data from the environment during its navigation, and (b) a tree-type memory structure to store this data in a simple manner to use it to anticipate upcoming events and to guide the robot's behavior in the absence of the external inputs.

Our experiments show that the proposed algorithm successfully built internal representations of the environment. These representations were capable of predicting upcoming concepts and of navigating the robot blindfolded in the environment, replacing missing sensory input.

The results in the 2nd experiment indicated that our algorithm had memorized the sequences of concepts found in the environment and the robot's behavior in each one. The robot's internal representation had captured the topology of the original environment. With such memory structure, the robot's previous knowledge could be recalled easily when it needed. In the final experiment, the robot indeed was able to navigate, to some degree, blindfolded using only its own internally built representation. The robot used its 'mind' to navigate from one concept to another by operating through a series of actions and situations that it learned. The robot's memory was not very good at predicting the real time needed for each concept, but neither can humans (Fig.1). This predicting error caused a little delay in the robot's movement.

Although some studies have reported on the issue of robot imaginations and anticipations in different ways (e.g., SLAM [12]), where the robot can use its sensorimotor representation in the brain to simulate its movement internally before the actual movement and to reason about its ability to perform the task in a short time and a safe manner, the robot, however, has been told the layout of the environment and/or the position of the targets in advance. We showed in this study, that the robot could build an environment's map and an appropriate sequence of events in it through its own experiences. The robot can then use this data to recover any missing or corrupted data and even plan its future movement within its internal representation before any actual move.

We believe that the work presented here illustrates some promising directions for further experimental investigations of vision-motor abstraction and for further developments of the synthetic phenomenology approach in general.

As a possible future set of experiments, it would be interesting to try to improve the learning algorithm of the second network by building a higher-level to control the prediction-layer operations in the prediction phase. This may decrease the learning time for newly created prediction-layers.

**Acknowledgments.** This work was supported by grants to KM from Japanese Society for Promotion of Sciences and from the University of Fukui.

## References

1. Gazzaniga, M.S.: The Cognitive Neurosciences III. MIT Press, Cambridge (2004)
2. Varela, F.J., Thompson, E., Rosch, E.: The embodied mind: cognitive science and human experience. MIT Press, Cambridge (1991)

3. Stening, J., Jacobsson, H., Ziemke, T.: Imagination and abstraction of sensorimotor flow: Towards a robot model. In: Chrisley, R., Clowes, R., Torrance, S. (eds.) Proceedings of the Symposium on Next Generation Approaches to Machine Consciousness, Hatfield, UK, pp. 50–58 (2005)
4. Linåker, F., Niklasson, L.: Extraction and inversion of abstract sensory flow representations. In: Proceedings of the Sixth international Conference on Simulation of Adaptive Behavior, From Animals to Animates, vol. 6, pp. 199–208. MIT Press, Cambridge (2000)
5. Nolfi, D.S., Tani, J.: Extracting regularities in space and time through a cascade of prediction networks: The case of a mobile robot navigating in a structured environment. Connection Science 11(2), 125–148 (1999)
6. Stening, J.: Exploring Internal Simulations of Perception in a Mobile Robot using Abstractions. Masters Thesis, School of Humanities and Informatics, University of Skövde, Sweden (2004)
7. Lee, D.N., Thompson, J.A.I.: Vision in action: the control of locomotion. In: Ingle, D., Goodale, M.A., Mansfield, R.J.W. (eds.) Analysis of Visual Behavior, pp. 411–433. MIT Press, Cambridge (1982)
8. Hesslow, G.: Conscious thought as simulation of behavior and perception. Trends in Cognitive Science 6(6), 242–247 (2002)
9. Taylor, T., Geva, S., Boles, W.W.: Monocular vision as a range sensor. In: Mohammadian, M. (ed.) Proceedings of International Conference on Computational Intelligence for Modeling, Control and Automation, pp. 566–575 (2004)
10. Alnajjar, F., Murase, K.: Self organization of spiking neural network that generates autonomous behavior in a real mobile robot. International Journal of Neural Systems 16(4), 229–239 (2006)
11. Alnajjar, F., Mohd Zin, I., Murase, K.: A Hierarchical Autonomous Robot Controller for Learning and Memory: Adaptation in Dynamic Environment. Adaptive Behavior 17(3), 179–196 (2009)
12. Vaughan, R., Zuluaga, M.: Use your illusion sensorimotor self-simulation allows complex agents to plan with incomplete self-knowledge. In: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J.C.T., Marocco, D., Meyer, J.-A., Miglino, O., Parisi, D. (eds.) SAB 2006. LNCS (LNAI), vol. 4095, pp. 298–309. Springer, Heidelberg (2006)

# Adaptively Coordinating Heterogeneous Robot Teams through Asynchronous Situated Coevolution

Abraham Prieto, Francisco Bellas, and Richard J. Duro

Integrated Group for Engineering Research
Universidade da Coruña, Spain
{abprieto,fran,richard}@udc.es

**Abstract.** Adapting to changing situations and objectives and selforganazing without a central controller in order to achieve an objective has become one of the main challenges in the design and operation of multirobot systems. The Asynchronous Situated Coevolution (ASiCO) algorithm has been successfully applied in surveillance tasks defined by just one global objective. In this paper we present the results obtained with ASiCO in more complex multirobot problems with several objectives that require a heterogeneous population of robot controllers that autonomously distribute the tasks. The paper focuses on the benefits of evolving an affinity coefficient that characterizes the individual genotypes.

**Keywords:** Coevolution, Adaptation, Multi-robot Systems, Coordination.

## 1 Introduction

This work is concerned with the application of a real time coevolutionary strategy, based on Watson et al's [1] Embodied Evolution (EE) concept that provides a means for groups of robots to selforganize and perform tasks in an efficient manner. The inspiration for this approach comes from the field of Artificial Life, but we have included some of the notions of utility functions found in the multiagent systems literature. In the original implementation of EE, the authors sought to establish a completely distributed evolutionary algorithm embodied in physical robots. Their approach was based on the hypothesis that a large number of robots could be used for the evaluation stage of an evolutionary process devoted to obtaining a controller for a particular task. This led to a set of ideas and design requirements that had to be taken into account, such as the fact that the evolutionary process had to be decentralized and thus the evaluations required for the determination of the fitness of an individual should take place directly within the individual in an embodied and localized manner, preferably on the physical robot itself. It is clear that this differs radically from other strategies found in the Evolutionary Robotics (ER) literature [2][3][4] where a centralized evolutionary algorithm is in charge of carrying out whatever operations are necessary using information from all the robots in a simulation (or even in some cases in real robots), and usually off-line, in order to obtain a controller that when instantiated on the robots produces the desired individual or group behavior.

Lately, a lot of effort has been devoted to the generation of coordinated behaviors for large (and sometimes not so large) groups of robots. Some authors have been

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 75–82, 2009.

working on the formalization of the problems [5] in order to produce hand crafted algorithms or controllers for particular tasks while others have concentrated on implementation issues [6][7][8]. However, many of these approaches are particular to a task (i.e. foraging or flocking) or environment, often using homogeneous sets of robots and/or controllers and do not provide a general framework for obtaining collective behaviors. Some work is beginning to appear where some problems are characterized and a study of the production of collective solutions is carried out in order to determine the most appropriate for each case. For an example see [9]. However, it is often ignored that the structure of most real world problems does not allow for an easy decomposition into subproblems , nor are they known beforehand and, consequently, the robot teams that solve them must arise in a joint manner. This is especially so in the case of complex dynamic problems where the environment or even the objectives change with time introducing a new level of complexity and a requirement for real time autonomous adaptation on the part of the robot teams.

The objective of this paper is to present the application of Asynchronous Situated Coevolution (ASiCo) as a valid distributed and adaptive strategy in order to allow for groups of robots to selforganize to perform tasks in an efficient manner. ASiCo draws inspiration from the main features of some Artificial Life based distributed evolutionary approaches but includes some ideas from the multiagent systems literature to provide a way to implement the objectives of the collective system through the creation of energy and interaction based utility distribution schemes. A first application to simple in multirobot surveillance tasks may be found in [10].

The following sections are devoted to providing a brief description of the algorithm as well as some important operators, such as embryonic based reproduction, required for the adaptation of the algorithm to real time operation on distributed multirobot systems. After this, we will show some results obtained from the application of this approach to the problem of obtaining controllers for a set of robots that must selforganize to perform a collective cleaning task. The objective is then made more difficult by having two different tasks that the robot team must concurrently perform, thus leading to the speciation of heterogeneous controller populations. This necessary speciation is made more efficient through the inclusion of an affinity term in the reproduction mechanism. Finally, some conclusions are presented.

## 2 A Brief Description of Asynchronous Situated Coevolution

The ASiCo algorithm is inspired on Artificial Life simulations in terms of the use of decentralized and asynchronous open-ended evolution. Unlike other bio-inspired approaches such as genetic algorithms in which selection and evaluation of the individuals is carried out in a centralized manner at regular processing intervals based on an objective function, this type of evolution is situated. This means that all of the interactions in the population are local and depend on spatial and temporal coincidence of the individuals in the simulation environment, implying an intrinsic decentralization. Consequently, reproduction, creation of new individuals or their elimination, is driven by events that may occur in the environment in a decentralized way.

This type of evolution has usually been employed for analysis purposes, this is, to study how a system evolves in an open-ended manner, and not really with an engineering objective in mind, and thus, there is no clear procedure to relate the global objective to be achieved with the local objectives of the agents that participate in the process. To this end, inspiration is taken from the studies of utility functions and their distribution among individuals in order to structure the energy dynamics of the environment to guide evolution to the objectives sought. Specifically, we have used the principled evaluation function selection procedure for evolving coordinated multirobot systems developed by Agogino and Tumer [11], which establishes a formal procedure to obtain the individual utility function from the global function. With this procedure, ASiCo open-ended evolution becomes an evolutionary optimization algorithm that provides a distributed solution by means of the whole population and not only by the best individual as in typical evolutionary algorithms.

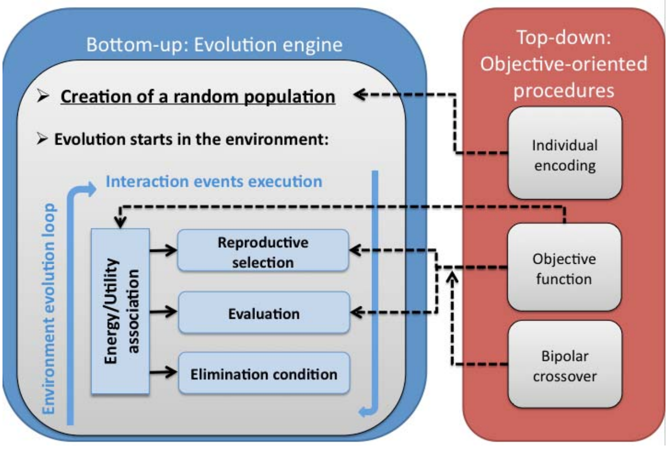

Bottom-up: Evolution engine
➢ Creation of a random population
➢ Evolution starts in the environment:
Interaction events execution
Environment evolution loop
Energy/Utility association
Reproductive selection
Evaluation
Elimination condition
Top-down: Objective-oriented procedures
Individual encoding
Objective function
Bipolar crossover

**Fig. 1.** Schematic representation of ASiCo structure

Fig. 1 displays a schematic representation of the algorithm's structure divided into two different parts, and the relations between these procedures and the processes carried out during evolution. On one hand (right block) we have the procedures that guide the evolution towards an objective. *Individual encoding* defines the solutions that can be generated, the *objective function* is established using the utility functions commented above, and, finally, *bipolar crossover* allows for the evolution of a heterogeneous population. On the other hand (left block) we have the *evolution engine*, which is based on the interactions among elements in the environment. After the *creation of a random population*, the execution of the interaction events occurs in a continuous loop modifying the state of the elements. In some conditions and based on energetic criteria for spatiotemporally coinciding individuals, the procedures that represent the evolution of the population (*selection*, *evaluation* and *elimination*) occur. The *energy/utility association* represents the energetic rules in the environment and affects these three procedures. Finally, the objective function defines the energetic criteria, the selection and the elimination.

Thus, ASiCo is an interaction driven algorithm. Interactions are a set of rules that make the state of the elements and individuals in the environment change in time due to particular events. This process is independent from the evolution of the population. Two elements are very relevant within ASiCo. On one hand the energy flow, that is, we have different rules that use the concept of energy to manage, through a bio-inspired strategy, the assignment of quality and the conditions to the creation and deletion of individuals in the population. On the other, reproductive selection is the set of rules that regulates the reproduction process. This selection process must be defined for each problem and is based on spatial interactions together with some energetic criteria. Specifically, it is usually performed by means of a tournament operator, which, in a typical evolutionary algorithm, randomly selects a number of individuals from the population for the reproduction. This centralized behavior is not possible here, so the tournament has been modified to be *asynchronous and decentralized* and, consequently, based on local interactions between the individuals.

A very important aspect of ASiCo is reproduction. This mechanism needs to be adapted to the objectives sought and in this paper we are interested in groups of real robots with a fixed number of individuals. Consequently, as we do not want real robots to "die" nor can we make robots "appear" in the environment, the reproduction mechanism has to contemplate the fact that the number of robots is fixed and still provide a way for the population to evolve in a distributed manner. Some authors have proposed different strategies, such as PGTA [1], which imply continuously modifying the genetic make up of the robots on line. This leads to instabilities in robot behaviors due to lack of evaluation time in the environment. Another approach, which is the one followed here, is to synchronize death with birth. In fact, as the robots are preset, we can only work with their controllers and thus, a death-birth process within a robot is just a change of its controller. To allow for evolutionary pressure within this process, we have designed a reproduction mechanism for fixed size populations called Embryo Based Reproduction (EBR).

The idea behind EBR is that each agent, carries, in addition to its own parameters, another set of parameters corresponding to its embryo and an associated pre-utility value for the embryo that estimates the utility of the agent generated from it. Thus, when a new agent is created, its embryo is generated as a mutation of the parent genotype with half of its energy. During the life of an agent, the embryo is modified whenever the agent meets another one and evaluates it positively, meaning that the average of the utility of the two parents is higher than the pre-utility of the current embryo. Finally, when the parent dies because it ran out of energy or time or for whatever other reason, the embryo substitutes the parent, that is, the control of the robotic unit is assumed by the embryo and a new embryo is generated within the robot. This way, we ensure that the size of the population remains constant and that the process takes place in an asynchronous and decentralized manner.

## 3 Experiments and Results

This section contains some of the results of applying ASiCo to the problem of obtaining controllers for a set of 20 preset robots that must carry out a task cooperatively. In the particular example we consider, the task has to do with cleaning an arbitrary area. Other authors have studied this problem using different algorithms [12][13], but our experience in multirobot surveillance tasks where ASiCO has provided better results as compared to typical evolutionary strategies [10], has led us to apply it in cleaning tasks too and thus analyze its capabilities in a typical testbed for multirobot systems.

The particular experiments we have developed consider the cleaning problem in a progressively more complex environment. It is made up of cells and each cell is characterized by a level of dirt that increases through a non-linear function while no robot cleans it. The dirt level goes to zero when the cell is cleaned and starts to increase again until it is re-cleaned by another robot. The final objective of the multirobot system is to keep the entire environment as clean as possible all the time. In terms of sensing, it is very primitive. A robot can detect how many robots are within the sensor reach (which is a constant for all the robots) in each of the four quadrants around it, the dirt level of the cell it is on and if a collision has occurred. In addition, each robot has a small memory that allows it to remember its position $n$ instants before. In terms of actions, all of the robots move at the same speed and the control system just provides a value for the angular velocity (rate of turn).

The control system is based on a RBFNN (Radial Basis Function Neural Network) whose parameters are encoded in the genotype of the robot together with the length of the memory. The inputs to this network correspond to the number of robots around it within the four quadrants, the module and angle of the vector relating its position to that $n$ instants before, the dirt level and its current position and the value of its collision sensor (whether a collision has occurred). The global utility is the sum of the dirt level in all the cells, and must be minimized. The individual utility is the sum of dirt levels in the cells in the instant the robot cleaned them.

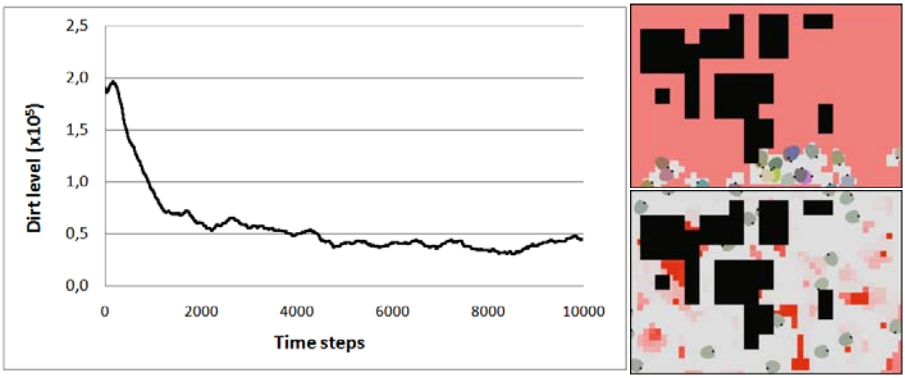


**Fig. 2.** Evolution of the global dirt level (left) and two screenshots of the simulation environment in simulation step 0 (top right) and simulation step 8000 (right bottom). Red intensity indicates dirt level.

Fig. 2 left shows the evolution of the global dirt level in an environment that is being controlled by a set of 20 robots as described above. As we can see, it decreases to a stable level in around 5000 simulation steps. To have a visual idea of the fulfillment of the task, the right images of Fig. 2 show two screenshots of the simulator. Different robot colors represent different genotypes and red intensity represents the dirt level. The top image shows the initial situation, where all the controllers are random. The bottom image corresponds to the state after around 8000 simulation steps. It is clear that the controllers of the robots have improved to the point of being able to obtain quite a low dirt level (an average dirt level of around 35000 units that, taking into account that the environment has 1584 cells, corresponds to 22 units per cell, consequently, every cell is being explored every 363 simulation steps). The fact that the whole area is monitored efficiently is a consequence of the emergence of a coordination strategy forced by the global utility requirements. Another interesting result that can be seen in Fig. 2 bottom right is that the robots tend towards a homogeneous genotype in this task (all the robots have the same color). This seems reasonable given the fact that the environment is relatively homogeneous in terms of requirements for the agents.

Once we have seen that ASiCO is able to solve the basic cleaning setup, we decided to study a more complex environment where the type of cleaning could be different depending on the zones. To simulate this situation we have designed an environment with two separate areas and provided the robots with a new sensor that detects if the area must be swept or vacuumed (we assume that all the robots can perform both tasks). The final result is shown in the screenshot of Fig. 3 left in simulation step 30000. As we can see, two different species have been created providing a very low global dirt level, represented in Fig. 3 right. This figure shows, in addition, the dirt level of the swept and vacuumed zones where we can appreciate initial fluctuations in the population towards robots that are capable of solving both tasks, and consequently, there is a sharp decrease in the dirt level in one zone or the other, but a high global dirt level. Around simulation step 6000, the population is divided into two different species and, consequently, both dirt levels decrease.

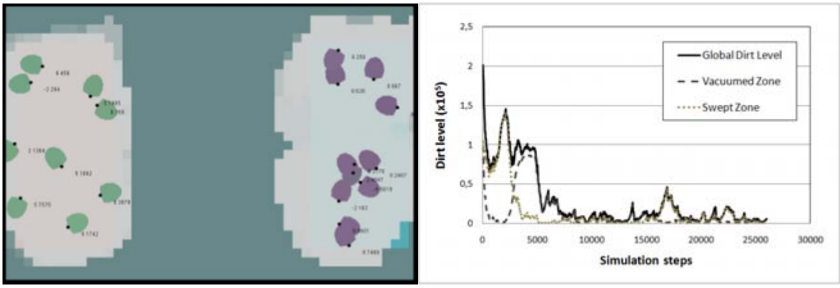


**Fig. 3.** Population of robots in simulation step 30000 when the cleaning problem is divided in two different tasks (left). Evolution of the global dirt level and that in both zones (right).

ASiCO automatically selforganizes the population into two different types of robots. This is a very interesting capability of the algorithm that can be used in a multirobot system design stage to automatically obtain the distribution of robots into types, in this case, with 11 vacuum robots and 9 sweep robots. This result is a consequence of the Bipolar crossovers, which does not favor the appearance of homogeneous genotypes, and the spatial separation of the two zones, that leads to robots executing small movements that keep them within a small area. Obviously, in a general problem, this spatial separation may not exist but the two tasks would be present, and ASiCO should provide different species too.

In Fig. 4 we have represented the environment created to study the emergence of species without such spatial separation of the zones. In the top left image, we show the final result obtained when we place together the two zones that in Fig. 3 left were separate. As we can see, the solution provided by ASiCO is a single species, so it seems that the basic implementation of the algorithm needs a spatial separation to obtain heterogeneous genotypes. To solve it, we have included a new coefficient in the genetic selection process, called affinity. This coefficient is calculated using the average difference of all the genes of two individuals, that is, it represents the genotypic distance of two individuals. During evolution, this coefficient is used to promote the selection and crossover of genotypically similar individuals, thus favoring the creation of species. In the case that the task does not require speciation, this coefficient does not affect the evolution.

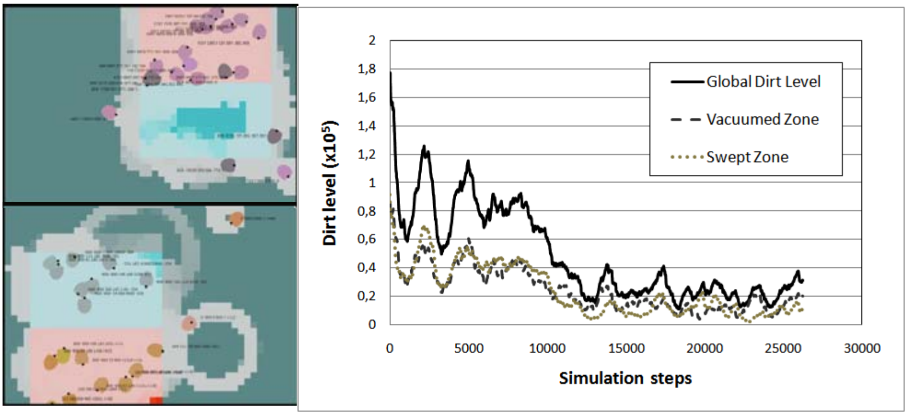


**Fig. 4.** Final multirobot system obtained when the two different areas that must be cleaned are placed together (left). The top image corresponds to the result obtained without affinity coefficient and the bottom one with it. Evolution of the dirt level (right).

After including the affinity coefficient, the result obtained is shown in Fig. 4 bottom left, where we can see two different species again. Fig. 4 right represents the evolution of the dirt level for the two zones and the global one. Now, the transient period that appeared in Fig. 3 right does not exist and all the dirt levels decrease from the beginning. With this improvement, the ASiCO algorithm is able to automatically provide a solution to the cleaning task with heterogeneous robots, specialized in sweeping or vacuuming.

## 4 Conclusions

The results presented in this paper provide an indication that Asynchronous Situated Coevolution (ASiCo) together with an embryonic like delayed reproduction mechanism (EBR) considering an affinity parameter is an effective approach for the introduction of real time evolution within robot coordination structures. The experiments here started from the premise that we had a fixed number of real robots and that the control strategy of the group had to arise in real time from their local interactions in a decentralized and asynchronous manner. It is clear from the results that, depending on the requirement of the problems, homogeneous or heterogeneous populations of controllers are obtained quite fast that allow the robot team to jointly achieve their objective and that when the objective or environment changes they adapt promptly to the new situations and objectives. The approach has been explored in a simulated environment that mimics the cleaning tasks that need to be carried out in real environments obtaining successful results.

**Acknowledgments.** This work was partially funded by the MEC of Spain through projects DEP2006-56158-C03-02 and DPI2006-15346-C03-01.

## References

1. Watson, R.A., Ficici, S.G., Pollack, J.B.: Embodied Evolution: Distributing an Evolutionary Algorithm in a Population of Robots. Robotics and Autonomous Systems 39(1), 1–18 (2002)
2. Harvey, I.: Artificial Evolution and Real Robots. In: Sugisaka, M. (ed.) Proceedings of the International Symposium on Artificial Life and Robotics (AROB), Beppu, Japan, pp. 138–141 (1996)
3. Mataric, M.J., Cliff, D.: Challenges in Evolving Controllers for Physical Robots. Journal of Robotics and Autonomous Systems 19(1), 67–83 (1996)
4. Floreano, D., Mondada, F.: Evolutionary Neurocontrollers for Autonomous Mobile Robots. Neural Networks 11(7-8), 1461–1478 (1998)
5. Kolling, A., Carpin, S.: Multi-robot Surveillance: an Improved Algorithm for the GRAPH-CLEAR Problem. In: Proc. 2008 IEEE International Conference on Robotics and Automation Pasadena, CA, USA, May 19-23, pp. 2360–2365 (2008)
6. Burgard, W., Moors, M., Stachniss, C., Schneider, F.E.: Coordinated multi-robot exploration Robotics. IEEE Transactions on Robotics 21(3), 376–386 (2005)
7. Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., Stewart, B.: Distributed Multirobot Exploration and Mapping. Proc. of the IEEE 94(7), 1325–1339 (2006)
8. Folgado, E., Rincón, M., Álvarez, J.R., Mira, J.: A Multi-robot Surveillance System Simulated in Gazebo. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2007. LNCS, vol. 4528, pp. 202–211. Springer, Heidelberg (2007)
9. Waibel, M., Keller, L., Floreano, D.: Genetic Team Composition and Level of Selection in the Evolution of Multi- Agent Systems. IEEE Transactions on Evolutionary Computation (to appear, 2009)
10. Schut, M.C., Haasdijk, E., Prieto, A.: Is Situated Evolution an Alternative for Classical Evolution? In: Proceedings CEC 2009, pp. 2971–2976 (2009)
11. Agogino, A., Tumer, K.: Efficient evaluation functions for evolving coordination. Evolutionary Computation 16(2), 257–288 (2008)
12. Chaomin, L., Yang, S.X., Stacey, D.A.: Real-time path planning with deadlock avoidance of multiple cleaning robots. Robotics and Automation 3, 4080–4085 (2003)
13. Wagner, I.A., Altshuler, Y., Yanovski, V., Bruckstein, A.M.: Cooperative Cleaners: A Study in Ant Robotics. The International Journal of Robotics Research 27(1), 127–151 (2008)

# RL-Based Memory Controller for Scalable Autonomous Systems

Osman Hassab Elgawi

School of Engineering, Tokyo Institute of Technology, Japan
osman@isl.titech.ac.jp

**Abstract.** This paper contributes on designing an autonomous system utilizing self-optimizing memory controller for non-Markovian reinforcement tasks. Instead of holistic search for the whole memory contents, the controller adopts associated feature analysis to produce the most likely relevant action from previous experiences. Actor-Critic (AC) learning is used to adaptively tuning the control parameters, while on-line variant of Random Forest (RF) learner is used as memory-capable to approximate the policy of Actor and the value function of Critic. Learning capability is experimentally examined through non-Markovian cart-pole balancing task. The result shows that the proposed controller acquired complex behaviors such as balancing two poles simultaneously and displays long-term planning.

## 1 Perceptual Aliasing and A Non-markovian Domain

As with the most real-world robot learning systems, the arising of perceptual aliasing, when the system has to scale up to a non-Markov setting or POMDP renders conventional reinforcement learning (RL) methods impracticable, raising an interests in heuristic methods without any world model (e.g., ‘memory approach’) that intrinsically and adaptively modifying the policy. In Memory-based systems the controller is unable to take optimal transitions unless it observed the past inputs, then the controller simultaneously solve the incomplete perception while maximizing discounted long-term reward. See [5,6] as an early practice attempts.

### 1.1 POMDP Formal Setting

The formal setting of POMDP or non-Markovian is: $\mathcal{P} = \langle \mathcal{M}, \mathcal{O}, \mathcal{Z} \rangle$ consist of

1. An MDP $M = \langle S, A, T, R \rangle$,
2. A set of possible observations $\mathcal{O}$, where $\mathcal{O}$ could constitute either a set of discrete observations or a set of real-value,
3. $\mathcal{Z}$, a probability density mapping state-observation combinations $S \times \mathcal{O}$ to a probability distribution, or in the case of discrete observations combinations $S \times \mathcal{O}$ to probabilities. In other words, $Z(s, o)$ yield the probability to observing $o$ in state $s$. So basically, a POMDP is like an MDP but with observations instead of direct state perception.

If a world model is available to the controller, it can easily calculate and update a *belief vector* $\boldsymbol{b_t} = \langle b_t(s_1), b_t(s_2), \cdots, b_t(s_N) \rangle$ over ‘hidden states’ at every time step $t$ by taking into a account the history trace $h = o_1, o_2, \cdots, o_{t-1}, o_t$.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 83–92, 2009.

### 1.2 Our Approach

The process of memorizing and scaling up could be lengthy if traditional memory scheduling processes are to be used. In order to speed up learning process and improve the convergence rate, a RL-controller is modeled as scheduler for our proposed self-optimizing adaptive memory controller (Fig.1). Rather than holistic search for the whole memory contents the model adopt associated feature analysis to successively memorize a newly experience (state-action pair) as an action of past experience. Actor-Critic (AC) learning is used to adaptively tuning the control parameters, while on-line variant of random forests (RF) [4] learner is used as memory-capable function approximator coupled with Intrinsically Motivated Reinforcement Learning (IMRL) reward function to approximate the policy of *actor* and the value function of *critic*. Some experimental results are presented as promising examples. It includes the non-Markovian cart-pole balancing and balancing two poles simultaneously.

At this point we would like to mention that M3 Computer Architecture Group at Cornell has proposed a similar work [7] to our current interest. They implement a RL-based memory controller with a different underlying RL implementation.

## 2 Self-optimizing Controller Architecture

One departing approach from manual ‘hard coding’ of behaviors is to let the controller build its own internal ‘behavior model’–‘on-the-fly’ by learning from past experience. Fig.1 illustrates the total view of our memory controller based on heuristic memory-based without any world model for solving varieties of non-Markovian reinforcement tasks. We briefly explain its components.

**Past experiences representation.** Sensory control inputs from environment would be stored at the next available empty memory location (chunk), or redundantly at several empty locations. In our memory implementation only the following parameters have to be specified by a designer: 1) The capacity of the memory, 2) A function which extracts features from its stored locations, 3) A predictor which provides relevant features of the current system state, and 4) A function which provides intrinsic rewards.

**Feature predictor.** Is utilized to produce associated feature for selective experience. This predictor was designed to predict multiple experiences in different situations. When the selective experience is predicted, the associated feature is converted to feature vector so the memory controller can handle it.

**Features Map.** The past experiences are mapped into multidimensional feature space using neighborhood component analysis, based on the Bellman error, or on the temporal difference (TD) error. In general this is done by choosing a set of features which approximate the states $S$ of the system. A function approximator (FA) must map these features into a value function $V^{\pi}$ for each state in the system. This generalizes learning over similar states and increases the speed of learning, but potentially introduces generalization error as the feature will not represent the state space exactly.

**Memory access.** The memory access scheduling is formulated as a RL agent whose goal is to learn automatically an optimal memory scheduling policy via interaction with the rest of the system. As can be seen in Fig.1 two scenarios are considered. In Fig.1a all the system parameters are *fully observable*, the agent can estimate $V^{\pi}$ for each state and use its actions (past experiences). In Fig.1b the system is *partially observable* [2,10], the agent does not know which state it is in due to sensor limitation, the agent updates its policy parameters directly. Our system is modeled as non-Markovian (non-MDP) process, where decision depends on last state-action, and the state transition $s_{t+1} = \delta(s_t, a_t)$ depend on arbitrary past state where the agent had visited. This transition is expressed by $Pr(s_t|s_{t-1}, a_{t-1}, s'_t, s''_t, \cdots)$, where $s_{t-1}$, $a_{t-1}$ are the previous state and action, and $t'$, $t''$ are arbitrary past time.

**Learning behaviors from past experience.** On each time step, a component of the TD learning algorithm, called the adaptive critic, is used to estimate the expected future reward of retaining various combinations of memory locations. The collection of memory locations show to have the highest accumulated rewards are more likely to be remembered. The amount of occasional intrinsic rewards received, a long with the estimates of the adaptive critic on this time step and on the previous time step, are used to compute the TD error–the change in expected future reward. This error signal also used to train the adaptive critic.

### 2.1 Conventional vs. Self-optimizing Memory Controller

Conventional manually designed memory controller suffers two major limitations in regard with scheduling process and generalization capacity. First, it can not anticipate the long-term planning of its scheduling decisions. Second, it lacks learning ability, as it can not generalize and use the experience obtained through scheduling decisions made in the past to act successfully in new system states. This rigidity and lack of adaptivity can lead to severe performance degradation in many applications, raising interest in self-optimizing memory controller with generalization capacity.

The proposed self-optimizing memory controller is a fully-parallel maximum-likelihood search engine for recalling the most relevant features in the memory of past. The memory controller considers the long-term planning of each available action. Unlike conventional memory controllers, self-optimizing memory controller has the following capabilities: 1) Utilizes experience learnt in previous system states to make good scheduling decisions in new, previously unobserved states, 2) Adapts to dynamically changing of the system parameters, and 3) Anticipates the long-term consequences of its scheduling decisions, and continuously optimizes its scheduling policy based on this anticipation.

No key words or pre-determined specified memory locations would be given for the stored experiences. Rather a parallel search for the memory contents would take place to recall the previously stored event that correlates with the current newly event. The controller handle the following tasks: (1) relate states and actions with the occasional

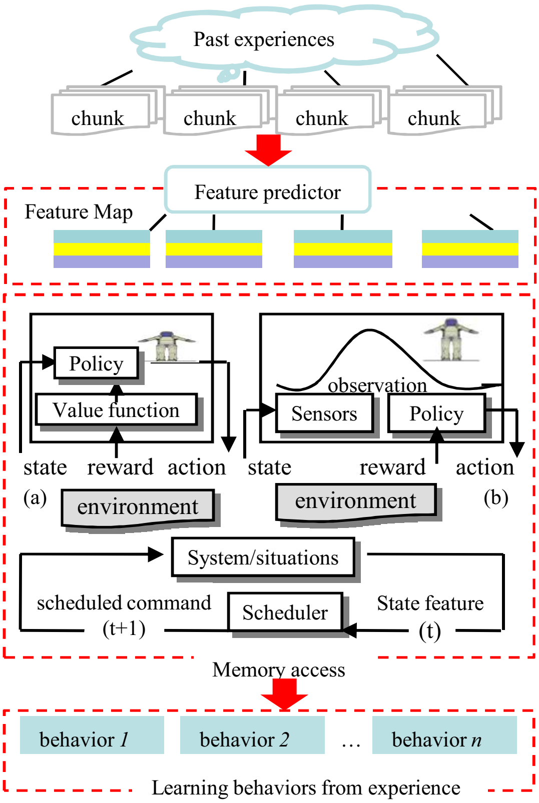


**Fig. 1.** Architecture of self-optimizing memory controller. The controller utilizes associated feature analysis to memorize complete non-Markovian reinforcement task as an action of past experience. The controller can acquired behaviors such as controlling objects, displays long-term planning and generalization capacity.

reward for long planning, (2) take the action that is estimated to provide the highest reward value at a given state, and (3) continuously update long-term reward values associated with state-action pairs, based on Intrinsically Motivated Reinforcement Learning (IMRL).

# 3 Memory Capable Function Approximation

## 3.1 Actor-Critic Learning

*Actor-critic*, a group of on-policy TD methods, separates the policy and the value function into independent memory structures. The policy structure, or *actor*, is used to decide which action to pick in each state. The estimate value function, or *adaptive critic*, determines whether the actions of the *actor* are to be rewarded or punished. The algorithms use these spare measures of performance to adopt an optimal behavior over time.

The adaptive critic maps its current state event onto an estimate of whether it will be rewarded. The mapping is learned from the past experience. If $s + 1$ is the situation that follows situation $s$ in time, this expected future reward may be written as:

$$V(s) = \gamma^0 r(s) + \gamma^1 V(s+1) + \cdots + \gamma^n V(s+n) \tag{1}$$

The value of the current situation, $V(s)$, is the sum of all the rewards we will receive over the next $n$ time steps. The rewards on each time step are "discounted" by factor, $\gamma$, in the range $[0, 1]$. Equation (1) may be rewritten in a recursive form:

$$V(s) = \gamma^0 r(s) + \gamma^1 V(s+1) = r(s) + \gamma V(s+1) \tag{2}$$

Any estimate of the value function that deviates from this equality is inaccurate, and the magnitude of inaccuracy is captured by the TD error:

$$\delta(s) = (r(s) + \gamma V(s+1) - V(s)) \tag{3}$$

Adopting these methods can save much computation for selecting optimal actions, due to utilizing separate memory for value function and policy.

### 3.2 Actor-Critic in Non-markovian Domain

Due to non-Markovian characteristics, explicit state information is not made available to the controller. Instead, the controller infers the state of its environment from a sequence of observation it receives, learns an optimal action by detecting certain past events, that associated with its current perception.
In particular, at time $t$, the error of the critic is give by

$$E_c(t) = \frac{1}{2}([r(t) + \gamma J(t)] - J(t-1))^2 \tag{4}$$

while the error of the actor is

$$E_a(t) = \frac{1}{2}(J(t)] - R^*)^2 \tag{5}$$

where $R^*$ is the optimal return, which is dependent on the problem definition. The expected return is expressed as the general cost function, $J(t)$, which is to be maximized by the controller. Specifically,

$$J(t) = r(t+1) + \gamma r(t+2) + \gamma^2 r(t+3) + \cdots \tag{6}$$

where $r(t)$ is the immediate reward and $\gamma$ is the time-discounting factor $0 \leq \gamma \leq 1$.

### 3.3 RF Memory-Enable for Optimal Learning

On-line RF has the characteristics of a simple structure, strong global approximation ability and a quick and easy training [4]. It has been used with TD learning for building a hybrid function approximator [11, 12]. Here, in order to reduce the demand of storage space and to improve learning efficiency, on-line RF is used to approximate policy function of *actor* and the value function of *critic* simultaneously. That is, the actor and the critic can share the input and the basis functions structure of the forest. In order for the actor to optimally select actions and the critic to correctly evaluate a state, both must able to accurately model the underlying dynamics of the controller's environment. The RF memory-enable is designed to eliminate this duplicated effort by embodying the actor and critic in a single structure. Let $RF_{Appro}$ represent a hybrid RF approximator that combines actor and critic. Given a state $s(t)$ and action $a(t)$, $RF_{Appro}$ is defined such that $RF_{Appro}(s(t), a(t)) = (J(t), a(t+1))$, where $J(t)$ is the estimated value of the given state-action pair, and $a(t+1)$ is the subsequent action to be taken by the controller. The error at the critic output is the squared Bellman error. However, the error at the action outputs is now determined by the gradient of the estimated value $J(t+1)$ w.r.t the action $a(t+1)$ selected by the on-line RF at time $t$. Specifically,

$$\begin{aligned} e_a(t) &= \alpha \nabla_{a(t+1) J(t+1)} \\ &= \alpha \left( \frac{\partial J(t+1)}{\partial a_1(t+1)}, \cdots, \frac{\partial J(t+1)}{\partial a_d(t+1)} \right) \end{aligned} \tag{7}$$

where $\alpha$ is a scaling constant and $d$ is the number of choices available at action $a$. Combining the error for each choice of the selected action, the overall actor error is

$$E_a(t) = \frac{1}{2} \left[ \sum_{i=1}^{d} e_{ai}^2(t) \right] \tag{8}$$

where $e_{ai}(t)$ is the choice of the action error gradient $e_a(t)$. In finding the gradient of the estimated value $J(t+1)$ w.r.t the previously selected action $a(t+1)$, the direction of change in action, which will improve the expected return at time step $t+1$, is obtained. Thus by incrementally improving actions in this manner, an optimal policy can be achieved. $E(t) = E_c(t) + E_a(t)$ define the reduced error for the entire on-line forest.

## 4 Experiment and Results

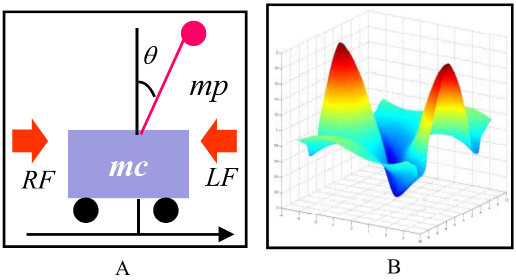
As discussed in previous sections, the proposed self-optimizing memory controller brings a number of preferable properties for learning different behaviors. In this section, we investigate its learning capability through a task of cart-pole balancing problem, designed with non-Markovian settings.

### 4.1 Related Work

The pole balancing algorithm has not been extensively modeled for non-MDP. Although a variation of Value and Policy Search (VAPS) algorithm [8] has been applied to this problem for the POMDP case [9], they have assumed that $x$ and $\theta$ are completely known or observable. If history for $x$ and $\theta$ are kept, their derivatives may be found, and all state variables will be revealed making the case a fully observable MDP. They have reached $500,000$ steps of iteration without failure. NeuroEvolution of Augmenting Topologies [13] and evolutionary computation [14], is another promising approach where recurrent neural networks are used to solve a harder balancing of two poles of different lengths, in both markovian and non-Markovian settings.

### 4.2 Non-markovian Cart Pole Balancing

As illustrated in Fig.2A, Cart-Pole balancing involves a vertical pole with a point-mass at its upper end installed on a cart, with the goal of balancing the pole when the cart moves by applying horizontal forces to the cart, which must not stray too far from its initial position. The state description for the controller consists of four continuous state variables, the angle $\theta$ (radial), and the speed of the pole $\acute{\phi} = \delta x/\delta t$ plus the position $x$ and speed of the cart $\acute{x} = \delta x/\delta t$. The two continuous actions set up for controller training and evaluation were RightForce (RF), (results in pushing the cart to the right), and LeftForce (LF), (results in pushing the cart left). At each time step $t$, the controller must only observe the $\theta$ and then takes appropriate action to balance the pole by learning from the past experience and the intrinsically rewards (Fig.2A). The optimal value function is shown in Fig.2B. A simulated sample run is shown in Fig.3. The controller could keep the pole balanced after about $4000$ steps.



A

B

**Fig. 2.** (A) Illustration of the non-Markov Cart-Pole balancing problem, where the angular velocity is not observing by the controller. (B) Optimal value function.

### 4.3 Non-markovian Two-Pole Balancing

Then we moved to a harder setting of this problem, balancing two poles simultaneously. Each pole has its own position and angular velocity, $\theta_1$ and $\dot{\theta}_1$ for the first pole and $\theta_2$ and $\dot{\theta}_2$ for the second pole respectively. The controller must balance the two poles without velocity information. In order to assist the feasibility of our approach to balance two poles simultaneously we compared with state-of-the-art methods. Table 1 reports the performance of our controller compared with traditional value function-based methods (including SARSA-CMAC, SARSA-CABA, and VAPS ) and policy search method (Q-MLP). Value function performance results are reported by [14], who used SARSA implementations by [15]. [14] implemented the Q-Learning system from which Q-Learning results are reported. It shows that our controller takes the minimal evaluations to balance the poles. With regard to CPU time (reported in seconds) we slightly fall short to Q-LMP. However, it interesting to observe that none of the value function approaches could handle this task in within the set of steps due to the memory constraint. The result also indicates that our memory controller stand as a promising method in solving this benchmark more successful than the traditional RL techniques.

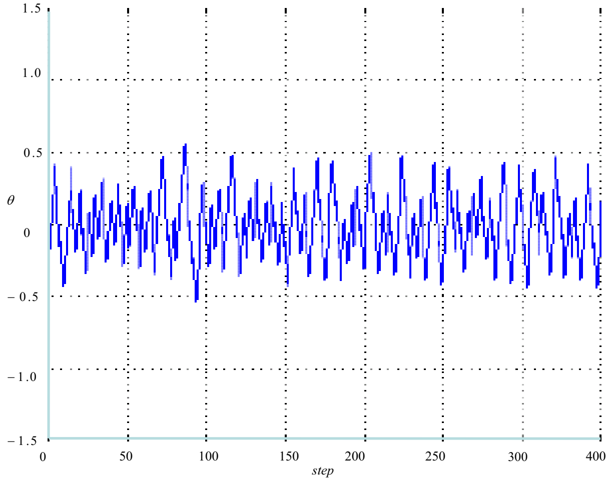


**Fig. 3.** A sample learning for balancing the pole

**Table 1.** Comparison of our result for balancing two carts simultaneously with state-of-the-art value function approaches and policy based methods

| | Method | Evaluation | CPU time (Seconds) |
|---|---|---|---|
| Value-function | SARSA-CMAC | Time Out | - |
| | SARSA-CABA | Time Out | - |
| | VAPS | Time Out | - |
| Policy-based | Q-MLP | 10,582 | 153 |
| Memory-based | Our | 8,900 | 300 |

## 5 Conclusions

In this paper we provide evidences that the robot control system will benefit from the inclusion of a self-optimizing memory controller. Our model avoids manual ‘hard coding’ of behaviors, modeled the memory controller as a RL agent learning from past experience. Results based on non-Markov Cart-Pole balancing indicate that our model can memorize complete non-Markovian sequential tasks and is able to produce behaviors that make the controlled system to behave desirably in the future. One of our future plans is to overcome the limited capacity of memory. In our current design the number of “chunks” that can be used is quite limited. Another future plan will be in designing intelligent mechanism for memory updating, and to experiment with different applications such as visual and speech recognition, and robot navigation.

## References

1. Chrisman, L.: Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In: Proc. Int'l. Conf. on AAAI, pp. 183–188 (1992)
2. Cassandra, A.R., Kaelbling, L.P., Littman, M.L.: Acting optimally in partially observable stochastic domains. In: Proc. Int'l. Conf. on AAAI, pp. 1023–1028 (1994)
3. Tsitsiklis, J.N., Van Roy, B.: Featured-based methods for large scale dynamic programming. Machine Learning 22, 59–94 (1996)
4. Hassab Elgawi, O.: Online Random Forests based on CorrFS and CorrBE. In: Proc. IEEE workshop on online classification, CVPR, pp. 1–7 (2008)
5. Jaakkola, T., Singh, S.P., Jordan, M.I.: Reinforcement learning algorithms for partially observable Markov decision. In: Advances in Neural Information Processing Systems 7, pp. 345–352. Morgan Kaufmann, San Francisco (1995)
6. Long-Ji, L., Mitchell, T.M.: Memory approaches to reinforcement learning in non-Markovian domains. Technical Report CMU-CS-92-138, School of Computer Science, Carnegie Mellon University (1992)
7. Ipek, E., Mutlu, O., Martinez, J.F., Caruana, R.: Self-Optimizing Memory Controllers: A Reinforcement Learning Approach. In: Intl. Symp. on Computer Architecture (ISCA), pp. 39–50 (2008)
8. Meuleau, N., Peshkin, L., Kim, K.-E., Kaelbling, L.P.: Learning finite-state controllers for partially observable environments. In: Proc of the 15th Int'l. Conf. on Uncertainty in Artificial Intelligence, pp. 427–436 (1999)
9. Peshkin, L., Meuleau, N., Kaelbling, L.P.: Learning policies with external memory. In: Bratko, I., Dzeroski, S. (eds.) Proc. of the 16th Int'l. Conf. on Machine Learning, pp. 307–314 (1999)
10. Kaelbling, L., Littman, M., Cassandra, A.: Planning and acting in partially observable stochastic domains. Artificial Intelligence 101, 99–134 (1998)
11. Hassab Elgawi, O.: Architecture of behavior-based Function Approximator for Adaptive Control. In: Köppen, M., et al. (eds.) ICONIP 2008, Part II. LNCS, vol. 5507, pp. 104–111. Springer, Heidelberg (2009)
12. Hassab Elgawi, O.: Random-TD Function Approximator. Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII) 13(2), 155–161 (2009)
13. Kenneth, O.S.: Efficient evolution of neural networks through complexification. Ph.D. Thesis; Department of Computer Sciences, The University of Texas at Austin. Technical Report AI-TR-04-314 (2004)
14. Gomez. F.: Robust non-linear control through neuroevolution. Ph.D. Thesis; Department of Computer Sciences, The University of Texas at Austin. Technical Report AI-TR-03-303 (2004)
15. Santamaria, J.C., Sutton, R.S., Ram, A.: Experiments with reinforcement learning in problems with continuous state and action spaces. Adaptive Behavior 6(2), 163–218 (1998)

## Appendix A: Pole-Balancing Learning Parameters

Below are the equations and parameters used for cart-pole balancing experiments [14]

**1. Pole-Balancing Equations**

The equations of motion for $N$ unjoined poles balanced on a single cart are

$$\ddot{x} = \frac{F - \mu_c sgn(\dot{x}) + \sum_{i=1}^{N} \tilde{F}_i}{M + \sum_{i=1}^{N} \tilde{m}_i}$$

$$\ddot{\theta}_i = -\frac{3}{4l_i}(\ddot{x}\cos\theta_i + g\sin\theta_i + \frac{\mu_{pi}\dot{\theta}_i}{m_i l_i}),$$

where $\tilde{F}_i$ is the effective force from the $i^{th}$ pole on the cart,

$$\tilde{F}_i = m_i l_i \dot{\theta}_i^2 \sin\theta_i + \frac{3}{4} m_i \cos\theta_i (\frac{\mu_{pi}\dot{\theta}_i}{m_i l_i} + g\sin\theta_i),$$

and $\tilde{m_i}$ is the effective mass of the $i^{th}$ pole,

$$\tilde{m_i} = m_i(1 - \frac{3}{4}\cos^2\theta_i).$$

**2. Pole-Balancing Learning Parameters**

**Table 2.** Parameters for the single pole & double pole problem

| | Parameters for the single pole | |
|---|---|---|
| Sym | Description | Value |
| $x$ | Position of cart on track | $[-2.4, 2.4]$m |
| $\theta$ | Angle of pole from vertical | $[-12, 12]$deg. |
| $F$ | Force applied to cart | $-10.10$N |
| $l$ | Half length of pole | 0.5m |
| $M$ | Mass of cart | 1.0kg |
| $m$ | Mass of pole | 0.1kg |
| | Parameters for double pole problem | Value |
| Sym | Description | Value |
| $x$ | Position of cart on track | $[-2.4, 2.4]$m |
| $\theta$ | Angle of pole from vertical | $[-36, 36]$deg. |
| $F$ | Force applied to cart | $-10.10$N |
| $l_i$ | Half length of $i^{th}$ pole | $l_1 = 0.5$m<br>$l_2 = 0.05$m |
| $M$ | Mass of cart | 1.0kg |
| $m_i$ | Mass of $i^{th}$ pole | $m_1 = 0.1$kg<br>$m_2 = 0.01$kg |
| $\mu_c$ | Coeff of friction on cart on track | 0.0005 |
| $\mu_p$ | Coeff of friction if $i^{th}$ pole's hinge | 0.0005 |

# A Semantic SLAM Model for Autonomous Mobile Robots Using Content Based Image Retrieval Techniques

Choon Ling Tan*, Simon Egerton*, and Velappa Ganapathy*

Monash University, Jalan Lagoon Selatan,
Bandar Sunway, 46150, Selangor Darul Ehsan,
Malaysia
cltan4@student.monash.edu,
simon.egerton@infotech.monash.edu.my,
velappa.ganapathy@eng.monash.edu.my
http://www.monash.edu.my

**Abstract.** Semantic approaches to conducting SLAM are still considered to be comparatively new compared to other methods. To this end, we introduce a new model for conducting SLAM on an autonomous mobile robot equipped with vision sensors. Our model consists of four separate stages, each with a specific goal at hand, namely: feature extraction, classification and storage, semantic analysis, and location resolving. This is the first time SLAM has been examined in this way and a set of planned experiments and benchmarks are also discussed which apply the proposed model to environments which are unknown and vary in their structure. Initial experiments are also included where images captured in different indoor locations are shown, along with the similarity scores of these images. Future work and experiments that are intended to be completed are then discussed.

**Keywords:** image patch, image segment, image signature, semantic SLAM, CBIR, Tamura texture.

## 1 Introduction

The problem of Simultaneous Localization and Mapping (SLAM) can be described as a situation where an autonomous robot – equipped with one or more low-level sensors – is placed within an unknown environment and is required to perform two tasks simultaneously: the first is known as localization where the robot estimates its current location in the environment relative to other "visible" objects, whereas the second is called mapping, where the robot records its own position in addition to any objects and obstacles within the current environment.

Having its origins from [20] where the issue of representing spatial information during the application of robotics is brought up, many traditional methods of conducting SLAM have been developed over the years. Such methods include those from Thrun [9], [22], [23], Davison [3], [19], and Little [6], [7]. While such methods of

* Choon Ling Tan and Simon Egerton are currently attached to the School of Information Technology, and Velappa Ganapathy is currently attached to the School of Engineering.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 93–106, 2009.

SLAM possess different implementations, they have the common feature of tracking topological information on a geometric level.

However, there have been several research efforts recently that have focused upon conducting SLAM on a semantic level [4], [8], [16] where high-level information is inferred from regular topological maps as suggested in [4]. This allows a greater amount of information pertaining to the environment (i.e. such as the navigability or the nature of certain areas) to be recorded, in addition to representing the presence of obstacles. For instance, in [17], certain occupied cells within an occupancy grid are labeled to indicate the presence of buildings, whereas in [18] and [8], certain regions of the map are classified according to different key objects that have been tracked.

Research into the field of semantic mapping is considered to be new. There are few papers that have been published regarding the subject and the majority of them are very recent. It is in the interest of contributing towards this field that we propose conducting semantic SLAM though content-based image retrieval (CBIR). Implementing CBIR to achieve a semantic form of SLAM is new in itself and provides advantages over traditional feature-based methods. For example, the semantic information has the potential to be queried by a human using natural language. Moreover, we believe such semantic methods have tenable biological parallels.

## 2 Model

We propose an approach of handling the SLAM problem by implementing CBIR with the aid of cameras. As seen in Fig. 1., the proposed model consists of four separate stages: (1) feature extraction, (2) classification and storage, (3) semantic analysis, and (4) location resolving. Each stage is composed of various sub-stages that contribute towards the overall goal of the stage. A description of each stage will be explained below.

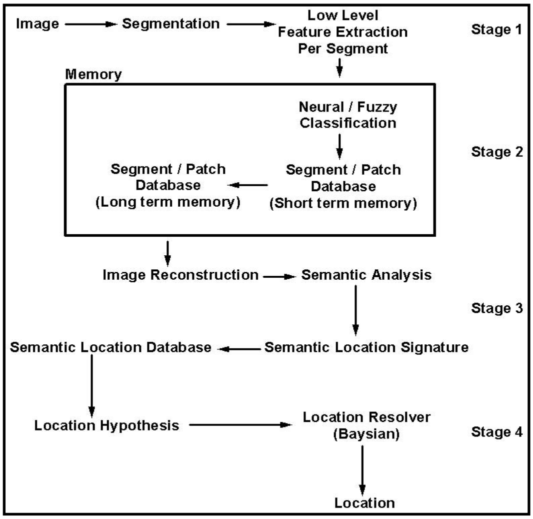


Image
Segmentation
Low Level Feature Extraction Per Segment
Stage 1
Memory
Neural / Fuzzy Classification
Stage 2
Segment / Patch Database (Long term memory)
Segment / Patch Database (Short term memory)
Image Reconstruction
Semantic Analysis
Stage 3
Semantic Location Database
Semantic Location Signature
Location Hypothesis
Location Resolver (Baysian)
Stage 4
Location

**Fig. 1.** The various stages within the proposed model

### 2.1 Stage 1 (Feature Extraction)

This stage begins with obtaining an image of the surrounding environment as input from vision sensors. This image is then segmented into 40 equally-sized portions, of which the process of feature extraction is conducted on each segment, or patch. The feature extraction in this last sub-stage is similar to that in [11] and [13] where the low-level Tamura texture features of coarseness, contrast and directionality [21] [1] are extracted as [12] has indicated the potential of implementing such features within the context of CBIR. In the event that such features prove to be inadequate in allowing sufficient variation in the latter stages of semantic analysis within the model, we will revise this stage accordingly to implement other features suitable to CBIR as documented in [5].

The features of coarseness, contrast and directionality are calculated by applying a procedure that is similar to the one implemented by [20], the following steps are performed to determine the coarseness value of a particular patch:

Step 1: For each pixel point (*x, y*) within the image patch, calculate the sum total over neighbourhoods where the sizes are powers of two (i.e. 1 * 1, 2 * 2, ..., 32 * 32). The sum total over the size $2^k$ neighbourhood at point (*x, y*) is

$$\mathrm{Sum}_{k,\,v}(x, y) = \sum_{i=x-2^{k-1}+1}^{x+2^{k-1}} \sum_{j=y}^{y+2^{k}-1} f(i,j) \tag{1}$$

for the vertical orientation, v, and

$$\mathrm{Sum}_{k,\,h}(x, y) = \sum_{i=x-2^{k}+1}^{x} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} f(i,j) \tag{2}$$

for the horizontal orientation, h, where f(i, j) is the gray-level pixel value at (x, y).

Step 2: For each and every pixel point of each value of k, calculate the differences between the sum total corresponding to pairs of non-overlapping, equal sized neighbourhoods in both vertical and horizontal orientations, where:

$$E_{k,\,v}(x, y) = (\mathrm{Sum}_{k,\,v}(x, y) - \mathrm{Sum}_{k,\,v}(x, y - 2^{k})) / 2^{2k} \tag{3}$$

for the vertical orientation, v, and

$$E_{k,\,h}(x, y) = (\mathrm{Sum}_{k,\,h}(x, y) - \mathrm{Sum}_{k,\,h}(x + 2^{k}, y)) / 2^{2k} \tag{4}$$

for the horizontal orientation, h.

Step 3: For each and every pixel point, analyse the output values as a result from the previous step, and select the best size, k, that gives the highest value:

$$S_{best}(x, y) = 2^{k} \tag{5}$$

where k maximizes E in either orientation:

$$E_{max} = \max\ (E_{1,\ h}, E_{1,\ v}, E_{2,\ h}, E_{2,\ v} \ldots, E_{5,\ h}, E_{5,\ v}) \tag{6}$$

Step 4: Finally, calculate the coarseness measure of the entire image patch by averaging all $S_{best}$ values:

$$F_{crs} = \frac{1}{m*n} \sum_{i=0}^{m} \sum_{j=0}^{n} S_{best}(i, j) \tag{7}$$

where m and n are the width and height of the image patch, respectively.

The method of measuring the value of contrast for any particular image patch is also explained in [20] with the following steps are performed:

Step 1: For each image patch, obtain a histogram of gray-level differences in order to, calculate the average gray-level value:

$$Avg_{image} = \ sum(\sum_{i=1}^{256} bin(i) * \frac{count(i)}{m*n}\ ) \tag{8}$$

where bin(i) is the i$^{th}$ bin value with a histogram count of count(i), and m and n are the width and height of the image patch, respectively.

Step 2: Calculate the fourth moment about the mean, μ4 with the following equation:

$$\mu 4 = sum((\sum_{i=1}^{256} bin(i)\ -\ Avg_{image})^{\ 4} * \frac{count(i)}{m*n})) \tag{9}$$

Step 3: Measure the amount of polarization by defining the kurtosis, $\alpha_4$ as

$$\alpha_4 = \mu_4 / \sigma^4 \tag{10}$$

where $\sigma^2$ is the variance, which is calculated as

$$\sigma^2 = sum((\sum_{i=1}^{256} bin(i)\ -\ Avg_{image})^{\ 2} * \frac{count(i)}{m*n})) \tag{11}$$

Step 4: Finally, combine σ and $\alpha_4$ to obtain the contrast measure with the equation

$$Fcon = \sigma / \alpha_4^{0.25} \tag{12}$$

[20] also provides the method of calculating the feature of directionality in which we implement as explained in the following steps:

Step 1: declare the following two 3 * 3 operators to aid in calculating the horizontal and vertical differences, ΔH and ΔV respectively, where

$$OP_H = \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \qquad OP_V = \begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}$$

Step 2: Calculate ΔH and ΔV for each gray-level pixel point *f*(*i, j*) in the image patch, where each 3 * 3 size neighbourhood with f(i, j) in the centre is multiplied with either OPH, or OPV, depending on the orientation, and then taking the sum total. These calculations for ΔH and ΔV can be summarized as:

$$\Delta H(x, y) = \mathrm{sum}(\sum_{i=x-1}^{x+1} \sum_{j=y-1}^{y+1} f(i, j) * OP_H) \qquad (13)$$

$$\Delta V(x, y) = \mathrm{sum}(\sum_{i=x-1}^{x+1} \sum_{j=y-1}^{y+1} f(i, j) * OP_V) \qquad (14)$$

Step 3: From ΔH and ΔV, we are able to obtain a magnitude ΔG, where

$$\Delta G = (|\Delta H| + |\Delta V|) / 2 \qquad (15)$$

and the local edge direction for each pixel point, θ(i, j), where

$$\theta(i, j) = \begin{cases} 0 & \text{iff } \Delta_H(i, j) = 0 \text{ and } \Delta_V(i, j) = 0 \\ \pi & \text{iff } \Delta_H(i, j) = 0 \text{ and } \Delta_V(i, j) > 0 \\ \tan^{-1}(\Delta_V(i, j) / \Delta_H(i, j)) + \frac{\pi}{2} & \end{cases} \qquad (16)$$

Step 4: Obtain the histogram HD by quantizing θ over bin values that range from 0 to π, as shown by an example in Figure 3. Following that, we apply a threshold process on to HD where any bin with a count value < 0.01 will have its respective count value be reset to 0. This is done in order to filter out the counting of unreliable directions that cannot be considered as edge points.

Step 5: The directionality is finally determined by calculating the sharpness of the peaks in HD. This is done by summing the peaks across the entire histogram, thusly:

$$Fdir = Fdir + \sum_{m=1}^{length(HD)} (\phi_m - \phi_p * 0.0001)^2 * H_D(m) \qquad (17)$$

where $\phi_m$ is the value of the m^th^ bin, $\phi_p$ is the bin value of the highest count (peak) within the histogram, and $H_D(m)$ is the count value of the mth bin.

From these features, a feature vector that is able to uniquely identify each patch is constructed, this is also known as an patch signature, and the 40 patch signatures within a single image constitute as a single image signature (shown in Fig. 2.), which is similar in concept to shape signatures used in [24]. Image signatures are required in order to provide some sort of semantic context to the location in which each batch of image patches was captured. This is done by comparing semantic descriptions that are extracted from the image signatures. Direct comparison between two sets of image signatures is not done; as it is possible that two similar images can generate different image signatures due to background noise (such as new objects appearing within the scene). Under such circumstances, comparison between image signatures would fail under typical fuzzy or Euclidean distance measures.

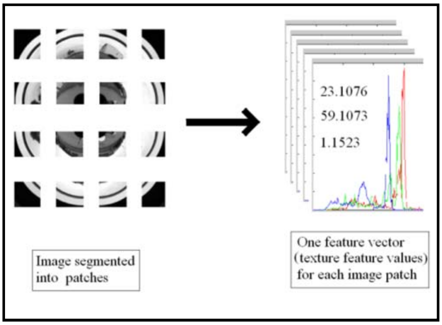

23.1076
59.1073
1.1523
Image segmented into patches
One feature vector (texture feature values) for each image patch

**Fig. 2.** Construction of an image signature

It is expected that by extracting semantic information, a match would be able to be produced, as new objects introduced in the scene may result in a different group of patches, but the unaffected patches will still be able to produce a semantic description that closely resembles its counterpart from the other image (without the offending object), and hence, generate a match.

From the fuzzy model of [13] and the variable weight assignment model as implemented by [14], it is suggested that matching high-level concepts between images is possible, but as with most CBIR techniques, relies on training before achieving a high success rate.

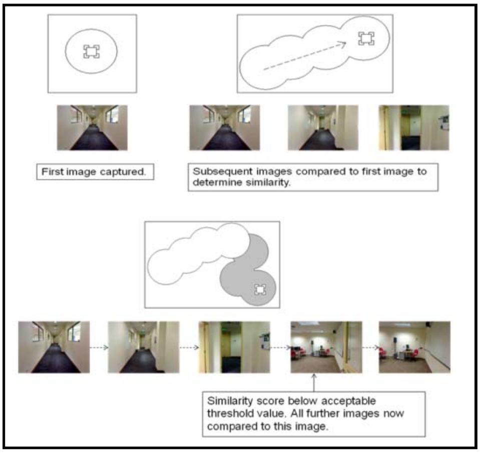
The Tamura texture features (i.e. coarseness, contrast, and directionality) which are extracted from each image patch have a single floating-point value attached to each of the features, which are calculated by applying a procedure that is similar to the one implemented by [21].

### 2.2 Stage 2 (Classification and Storage)

After the process of constructing patch signatures is completed, they are then presented to a fuzzy/neural network classifier, or some other appropriate self-organizing network. This is an important factor to consider, as the classifier should not be limited to a fixed number of classes, but grow accordingly instead as the variance in signatures demands an increasing number of classes.

Each class is determined by a <patch signature, patch> pairing, which is required during image reconstruction in the next stage. Note that a graphical, visual reconstruction of images is not necessary at this development stage, as the proposed semantic analysis would work equally as well on class labels generated from the self-organizing neural network. Such an action may still prove to be useful in later developments in order to aid in enhancing semantic descriptions (i.e. using object recognition or further CBIR techniques).

Thus, the classifier acts as the short-term memory portion of this stage, while <patch signature, patch> pairs being copied into long-term memory whenever a semantic description uses that pairing. However, it is possible for the classifier to generate classes that will never be used. Through this method, a robot traversing through an environment will classify captured images according to Fig. 3.



First image captured.

Subsequent images compared to first image to determine similarity.

Similarity score below acceptable threshold value. All further images now compared to this image.

**Fig. 3.** Classification of images

The input variables consist of the three Tamura texture features of coarseness, contrast and directionality, each with 2 trapezoidal-shaped membership functions. A single output variable with 2 trapezoidal-shaped membership functions is determined after the input variables are processed through a fuzzy rule set in order to determine the final similarity score of any particular image. This can be seen in Fig. 4. below.

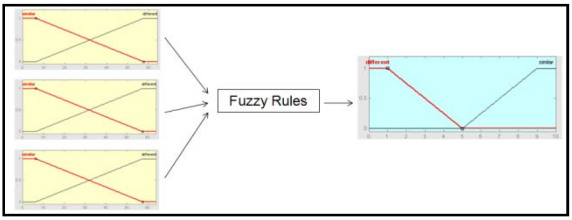
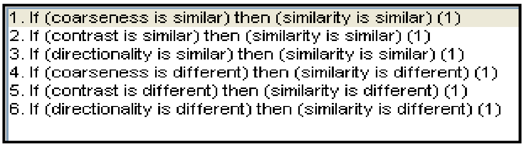


**Fig. 4.** Input/Output variables of the fuzzy inference system

The fuzzy rule set currently consists of 6 rules where each input variable-membership function tuple is mapped to the corresponding membership function in the output variable, as seen in Fig. 5.



1. If (coarseness is similar) then (similarity is similar) (1)
2. If (contrast is similar) then (similarity is similar) (1)
3. If (directionality is similar) then (similarity is similar) (1)
4. If (coarseness is different) then (similarity is different) (1)
5. If (contrast is different) then (similarity is different) (1)
6. If (directionality is different) then (similarity is different) (1)

**Fig. 5.** Fuzzy rules of the inference system

### 2.3 Stage 3 (Semantic Analysis)

Once patch signatures have been classified into various groups, each and every captured image is to be reconstructed through the use of only the unique patch signatures that have been stored within the database. Following this, reconstituted images are then analyzed in order to determine the relationship between image patches. Each cluster of patches that are within the same classification is given labels denoting the semantic relationship with other patch clusters as well as its current location within the entire image. This can be seen in Fig. 6, where the patch cluster of C1 is to the left of the patch cluster of C2.

| | | | | | |
|---|---|---|---|---|---|
| | C1 | C1 | C2 | | |
| | C1 | C1 | C2 | | |
| | | | C2 | | |
| | | | C2 | | |
| | | | | | |
| | | | | | |

**Fig. 6.** Relative locations between patch clusters

Therefore, the cluster of C1 is to be annotated with the relationship labels of:

Right(C1) = C2
Location(C1) = top left

In order to provide a certain degree of flexibility in regards to relationships between clusters, we attach high-level descriptors to them, rather than noting the specific patch indices that constitute a cluster. For instance, if cluster C1 were to shift left, the labels denoting the relationship between C1 and C2 would still hold, whereas that might not necessarily be true if specific patch indices were recorded.

The aggregate of all labels related to one cluster is considered to be the semantic location signature for that cluster. There also exists the possibility of detecting high-level features that are present within the captured images such as object recognition [18] to further contribute towards the amount of semantic data present within each semantic location signature.

As an example, consider the image in Fig. 7(a), which has 5 potential semantic areas of interest that can be treated as classes. These areas are divided within the regions as shown in Fig. 7(b) where C1 to C5 are the counter, ceiling, door, wall/railing, and floor, respectively. Therefore, the reconstituted counterpart to this image would be as seen in Fig. 7(c) in the next page.

The semantic descriptions for each and every cluster (and hence, the entire reconstructed image) is then entered into a semantic descriptor database. This database is responsible for comparing existing descriptors within in against their incoming counterparts to determine if a match is possible. Towards this end, a semantic reasoning system with an implementation similar to that of [2] with Research Cyc as the knowledge base of choice.

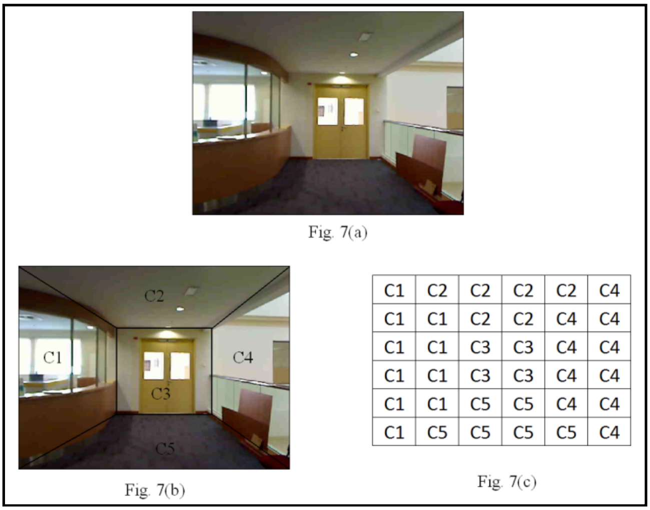


Fig. 7(a)

Fig. 7(b)

| | | | | | |
|---|---|---|---|---|---|
| C1 | C2 | C2 | C2 | C2 | C4 |
| C1 | C1 | C2 | C2 | C4 | C4 |
| C1 | C1 | C3 | C3 | C4 | C4 |
| C1 | C1 | C3 | C3 | C4 | C4 |
| C1 | C1 | C5 | C5 | C4 | C4 |
| C1 | C5 | C5 | C5 | C5 | C4 |

Fig. 7(c)

**Fig. 7.** Reconstruction of an image

## 2.4 Stage 4 (Location Resolving)

As a result of the previous stage, a set of location hypothesis (i.e. a set of locations each labeled with a probability) is generated. Such sets are generally caused by noise and perceptual aliasing (a situation where different locations appear similar to each other). Therefore, this final stage of resolving which location the robot is currently at is required, as simply selecting the location with the highest probability is naïve.

Bayesian reasoning opens up the possibility of determining the most likely location the robot is currently at by considering where the robot assumes its current location is, along with a new set of location hypothesis. However, a common problem that arises from this is method is the issue of location, or orientation independence. As an example, it can be difficult to recognize that a previous location has been revisited as it was viewed from a significantly different angle. One solution towards this problem is to implement a panoramic image sensor, which can be financially prohibitive and not easily available. Therefore, we attempt to overcome the aforementioned issue through another method, where locations are semantically related to each other by tracking either unique patches, or a unique ordering of patches.

## 3 Experiments

Three preliminary experiments have already been conducted in the MATLAB environment, where images with a resolution of 640 * 480 pixels are segmented in a similar manner to that in [10]. However, in order to obtain a meaningful amount of semantic data, each image patch consists of 96 * 80 pixels, resulting in 40 separate image patches per image. The size of image patches is determined through the use of the bestblk command within the MATLAB environment, which specifies the optimal patch size for processing.

The results of our experiments are shown in Fig. 8., where the 20th patch of 3 separate images are analyzed and their respective color and texture features are extracted. It is expected that the differences between image patches of separate images (i.e. as a result of their feature comparisons) can be transposed to a higher, semantic level, which also implies differences between entire images, as an entire image can be said to be a larger version of an image patch with its signature composed of multiple, smaller sub-signatures.

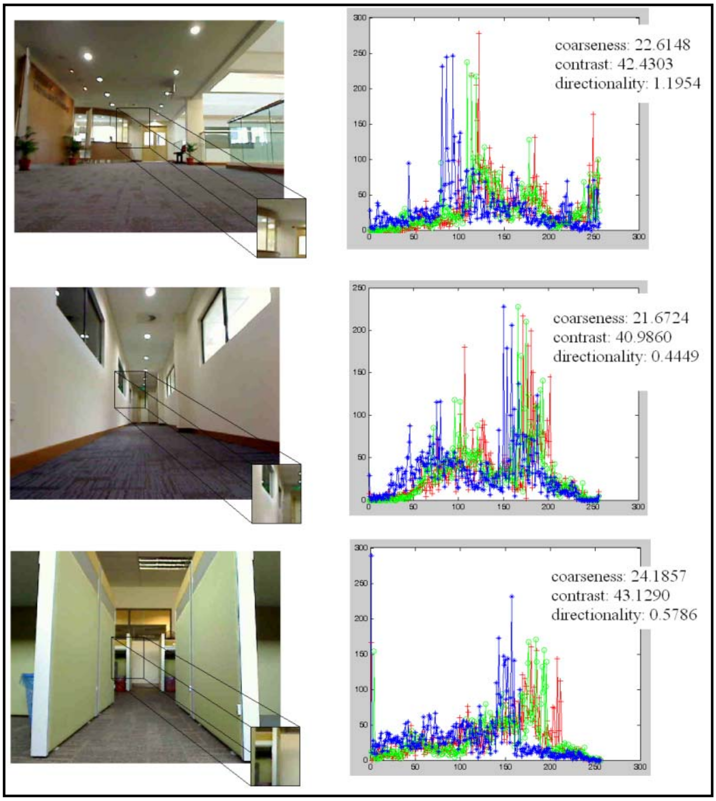
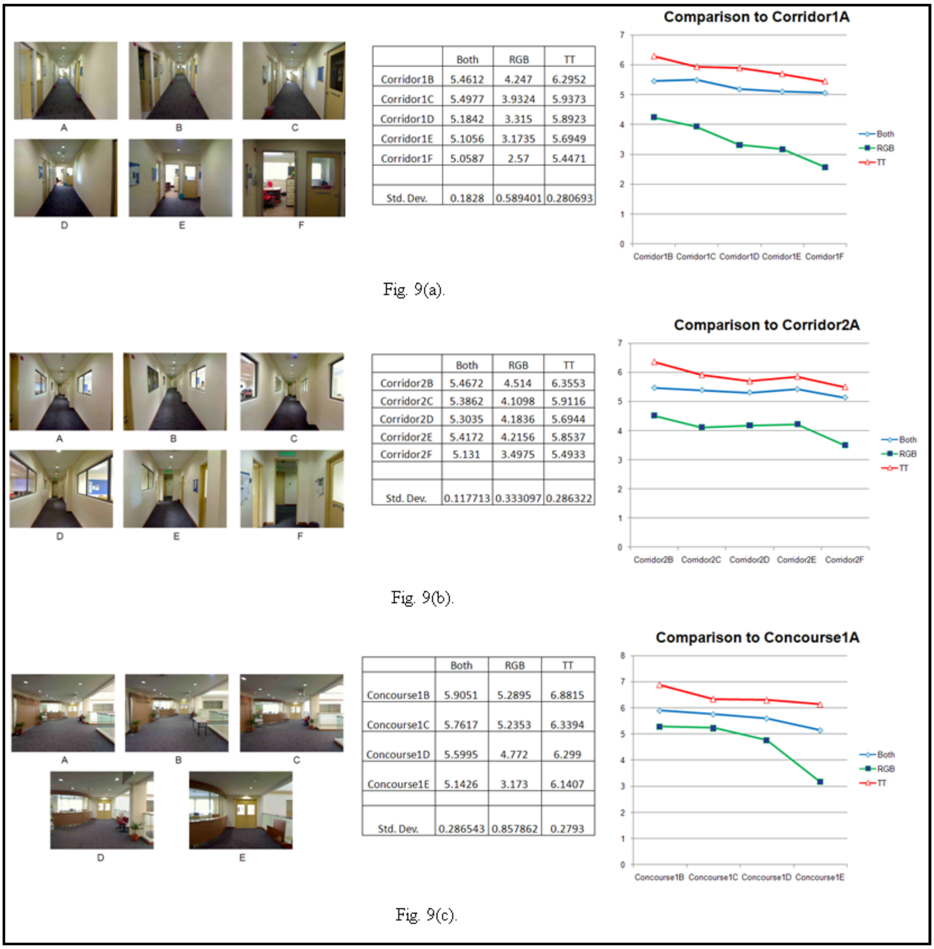


**Fig. 8.** Images of an open concourse area (top), a corridor (middle), and a cubicle room (bottom), with the extracted feature information of the indicated image patch region

These results appear to be encouraging. While the color histograms for each image patch have a unique distribution curve, there is also the possibility that the Tamura texture feature values also tend to favor a specific image-location pairing, with each image possessing a specific feature value that is distinct from the others. In this case, the image patch for the open concourse area has a high directionality compared to the others, while the cubicle room image patch has a somewhat higher coarseness.

Following this, the next experiment applies the fuzzy logic system from [15] on to the image signature vector to determine if Tamura textures alone are able to detect differences between images in the same capacity as RGB histograms. Experiments with the RGB histogram and Tamura texture feature components were conducted to



| | Both | RGB | TT |
|---|---|---|---|
| Corridor1B | 5.4612 | 4.247 | 6.2952 |
| Corridor1C | 5.4977 | 3.9324 | 5.9373 |
| Corridor1D | 5.1842 | 3.315 | 5.8923 |
| Corridor1E | 5.1056 | 3.1735 | 5.6949 |
| Corridor1F | 5.0587 | 2.57 | 5.4471 |
| | | | |
| Std. Dev. | 0.1828 | 0.589401 | 0.280693 |

Fig. 9(a).

| | Both | RGB | TT |
|---|---|---|---|
| Corridor2B | 5.4672 | 4.514 | 6.3553 |
| Corridor2C | 5.3862 | 4.1098 | 5.9116 |
| Corridor2D | 5.3035 | 4.1836 | 5.6944 |
| Corridor2E | 5.4172 | 4.2156 | 5.8537 |
| Corridor2F | 5.131 | 3.4975 | 5.4933 |
| | | | |
| Std. Dev. | 0.117713 | 0.333097 | 0.286322 |

Fig. 9(b).

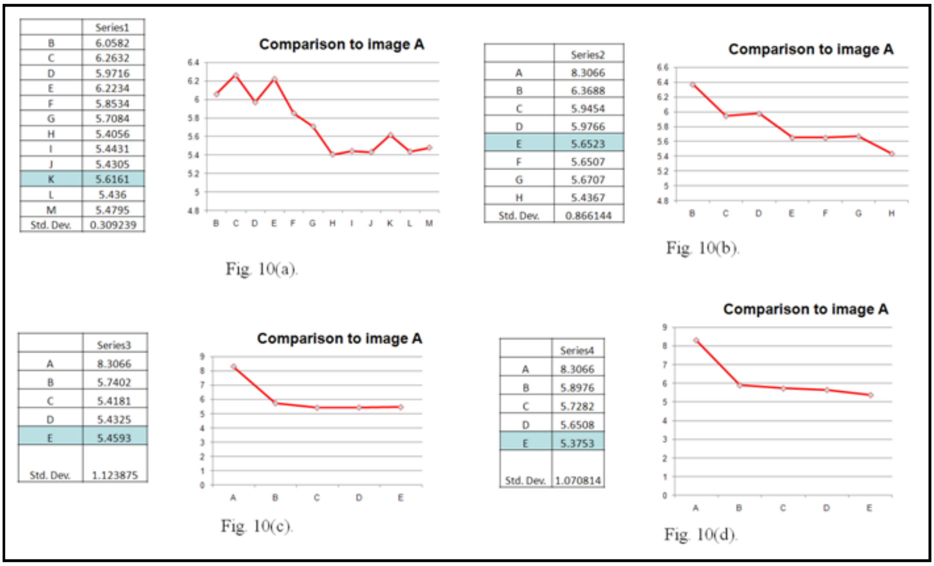
| | Both | RGB | TT |
|---|---|---|---|
| Concourse1B | 5.9051 | 5.2895 | 6.8815 |
| Concourse1C | 5.7617 | 5.2353 | 6.3394 |
| Concourse1D | 5.5995 | 4.772 | 6.299 |
| Concourse1E | 5.1426 | 3.173 | 6.1407 |
| | | | |
| Std. Dev. | 0.286543 | 0.857862 | 0.2793 |

Fig. 9(c).

**Fig. 9.** The image sets of Corridor1 (a), Corridor 2 (b) and Concourse (c), with their respective results

determine the degree of influence of each image signature component in fuzzy logic classification. Data for this experiment consists of 3 separate locations, 2 corridor sets and 1 concourse set, which can be seen in Fig. 9. The compiled results indicate a high possibility that Tamura texture features alone are able to be implemented in the proposed model. With only one exception (Corridor2E), the values of all image signatures follow a decreasing trend as the robot traverses the various environments selected for the experiment.

The goal of the final experiment is to determine the similarity score value which indicates a change in area (i.e. from a corridor to a room) during the classification process. To this end, 4 separate image sequences have been recorded, each consisting of 2 separate areas. The results are shown in Fig. 10., where the shaded letters indicate that the location of that image is semantically different than its predecessors (i.e. in Fig. 10(a), image A – J consists of a corridor area, whereas K – M consists of a room area). These results indicate that threshold value in similarity is around roughly 5.37 – 5.65. However, there have been several instances where images belonging within the same area have values below this threshold, such as in images H – J in Fig. 10(a). Therefore, further refinement towards the model is required before attempting to proceed into further stages within the proposed model.



| | Series1 |
|---|---|
| B | 6.0582 |
| C | 6.2632 |
| D | 5.9716 |
| E | 6.2234 |
| F | 5.8534 |
| G | 5.7084 |
| H | 5.4056 |
| I | 5.4431 |
| J | 5.4305 |
| K | 5.6161 |
| L | 5.436 |
| M | 5.4795 |
| Std. Dev. | 0.309239 |

Fig. 10(a).

| | Series2 |
|---|---|
| A | 8.3066 |
| B | 6.3688 |
| C | 5.9454 |
| D | 5.9766 |
| E | 5.6523 |
| F | 5.6507 |
| G | 5.6707 |
| H | 5.4367 |
| Std. Dev. | 0.866144 |

Fig. 10(b).

| | Series3 |
|---|---|
| A | 8.3066 |
| B | 5.7402 |
| C | 5.4181 |
| D | 5.4325 |
| E | 5.4593 |
| Std. Dev. | 1.123875 |

Fig. 10(c).

| | Series4 |
|---|---|
| A | 8.3066 |
| B | 5.8976 |
| C | 5.7282 |
| D | 5.6508 |
| E | 5.3753 |
| Std. Dev. | 1.070814 |

Fig. 10(d).

**Fig. 9.** Results for dual-area experiments

## 4 Conclusion

In this paper, we have presented a semantic based SLAM model. This model is based on four separate stages with the goals of feature extraction, classification and storage, semantic analysis, and location resolving. We also describe three preliminary experiments which demonstrate that implementation of Tamura texture features (i.e. coarseness, contrast, and directionality) alone are capable of producing noticeable differences in similarity scores when supplied with a series of images as a mobile robot traverses an environment. While further development is required for Stages 3 and 4 of the proposed model, we see our model as potential contribution towards semantic, and especially ontologically based methods that play a key role in future developments of both SLAM and generalized mapping models.

## References

1. Chiu, C.Y., Lin, H.C., Yang, S.N.: A Fuzzy Logic CBIR System. In: 12th IEEE International Conference on Fuzzy Systems, FUZZ 2003, May 28, vol. 2, pp. 1171–1176 (2003)
2. Daoutis, M., Coradeschi, S., Loutfi, A.: Integrating Common Sense in Physically Embedded Intelligent Systems. In: Proceedings of the 5th International Conference on Intelligent Environments, vol. 2, pp. 212–219 (2009)
3. Davison, A.: Real-Time Simultaneous Localization and Mapping with a Single Camera. In: Proceedings of the Ninth International Conference on Computer Vision ICCV 2003, Nice, France, pp. 1403–1410 (2003)
4. Dellaert, F., Bruemmer, D.: Semantic SLAM for Collaborative Cognitive Workspaces. In: Presented at the AAAI Fall Symposium Series, Arlington, VA, USA (2004)
5. Deselaers, T., Keysers, D., Ney, K.: Features for Image Retrieval: An Experimental Comparison. Information Retrieval 11(2), 77–107 (2008)
6. Elinas, P., Little, J.J.: Stereo Vision SLAM: Near Real-Time Learning of 3D Point-Landmark and 2D Occupancy-Grid Maps Using Particle Filters. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2007 (2007) (to be published)
7. Elinas, P., Sim, R., Little, J.J.: SLAM: Stereo Vision SLAM Using the Rao-Blackwellised Particle Filter and a Novel Mixture Proposal Distribution. In: Proc. 2006 IEEE International Conference on Robotics and Automation (ICRA 2006), pp. 1564–1570 (2006)
8. Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernandez-Madrigal, J., Gonzalez, J.: Multi-hierarchical Semantic Maps For Mobile Robotics. In: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS 2005), Edmonton, CA, USA, pp. 3492–3497 (2005)
9. Hahnel, D., Thrun, S., Wegbreit, B., Burgard, W.: Towards Lazy Data Association in SLAM. In: Proceedings of the 11th International Symposium of Robotics Research (ISRR 2003), Sienna, Italy (2003)
10. Jhanwar, N., Chaudhuri, S., Seetharaman, G., Zavidovique, B.: Content Based Image Retrieval Using Motif Cooccurrence Matrix. Image and Vision Computing 22(14), 1211–1220 (2004)
11. Jonsgård, O.A.F.: Improvements on Colour Histogram-based CBIR. Gjøvik University College, Norway (2005)
12. Kulkarni, S., Verma, B.: Fuzzy Logic based Texture Queries for CBIR. In: Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2003), pp. 223–228 (2003)
13. Lakdashti, A., Moin, M.S., Badie, K.: Semantic-Based Image Retrieval: A Fuzzy Modeling Approach. In: IEEE/ACS International Conference on Computer Systems and Applications (AICCSA 2008), Doha, Qatar, pp. 575–581 (2008)

14. Liu, P., Jia, K., Wang, Z., Lv, Z.: A New And Effective Image Retrieval Method Based On Combined Features. In: Fourth International Conference on Image and Graphics (ICIG 2007), Sichuan, China, pp. 786–790 (2007)
15. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Monocular Vision Based SLAM for Mobile Robots. In: Proceedings of the 18th International Conference on Pattern Recognition, vol. 3, pp. 1027–1031 (2006)
16. Nuchter, A., Surmann, H., Lingemann, K., Hertzberg, J.: Semantic Scene Analysis Of Scanned 3D Indoor Environments. In: Proceedings of the 8th International Fall Workshop Vision, Modeling, and Visualization 2003 (VMV 2003), Munich, Germany, pp. 215–222 (2003)
17. Persson, M., Duckett, T., Valgren, C., Lilienthal, A.: Probabilistic Semantic Mapping with a Virtual Sensor for Building Nature Detection. In: Proc. IEEE Int. Symp. on Computational Intelligence in Robotics and Automation (CIRA 2007), pp. 236–242 (2007)
18. Rottmann, A., Mozoz, O.M., Stachniss, C., Burgard, W.: Semantic Place Classification of Indoor Environments with Mobile Robots using Boosting. In: Proc. Nat. Conf. Artif. Intell (AAAI), pp. 1306–1311 (2005)
19. Smith, P., Reid, I., Davison, A.: Real-Time Monocular SLAM with Straight Lines. In: British Machine Vision Conference, vol. 1, pp. 17–26 (2006)
20. Smith, R., Self, M., Cheeseman, P.: Estimating Uncertain Spatial Relationships in Robotics. In: Autonomous Robot Vehicles, pp. 167–193. Springer, Heidelberg (1990)
21. Tamura, H., Mori, S., Yamawaki, T.: Textural Features Corresponding to Visual Perception. IEEE Transactions on Systems, Man and Cybernetics 8(6), 460–473 (1978)
22. Thrun, S.: Robotic Mapping A Survey. In: Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann, San Francisco (2002)
23. Thrun, S., Thayer, S., Whittaker, W., Baker, C., Burgard, W., Ferguson, D., Hahnel, D., Montemerlo, D., Morris, A., Omohundro, Z., Reverte, C., Whittaker, W.: Autonomous Exploration and Mapping of Abandoned Mines. IEEE Robotics and Automation Magazine 11(4), 79–91 (2004)
24. Zhang, D., Lu, G.: A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures. In: Proc. of International Conference on Intelligent Multimedia and Distance Education (ICIMADE 2001), Fargo, ND, USA, June 1, pp. 1–9 (2001)

# Parameter Estimation Using a SCE Strategy

Pengfei Li, Hesheng Tang, and Zhaoliang Wang

Research Institute of Structural Engineering and Disaster Reduction, Tongji University,
Shanghai 200092 PRC

**Abstract.** As a novel evolutionary computation technique, shuffled complex evolution (SCE) has attracted much attention and wide applications for solving complex optimization problems in different fields. This paper utilizes SCE strategy to estimate parameters of structural systems, which could be formulated as a multi-modal optimization problem with high dimension. Simulation results for identifying the parameters of a structural system under four conditions including limited output data, noise polluted signals, and no prior knowledge of mass, damping, or stiffness are presented to demonstrate the effectiveness of the proposed method.

## 1 Introduction

In system identification, considerable efforts have been invested in developing methods for identification of system models and their parameters. For civil-engineering systems, limited progress has been made with analytical methods for complexity and incomplete prior information. Instead, some successes have been achieved with various intelligent optimization algorithms. Evolution strategy (ES) algorithms have been presented for the identification of multiple degree of freedom (DOF) systems [1]. Perry et al. [2] have presented a modified GA to identify structural systems.

As a novel evolutionary computation technique, Shuffled complex evolution (SCE) has attracted much attention and wide applications, owing to exactness of solution, easy implementation and quick convergence [3].

The SCE method is based on a synthesis of four concepts that have proved successful for global optimization: (a) combination of probabilistic and deterministic approaches; (b) clustering; (c) systematic evolution of a complex of points spanning the space, in the direction of global improvement; and (d) competitive evolution.

In the field of structural engineering, especially system identification, SCE still has not been applied widely. Meanwhile, although numerous traditional approaches in literature tackled the problem of system identification in the field of civil engineering, it is still difficult for these approaches to extract the physical characteristics of the system like mass, damping, or stiffness in a structural system unless some of these are assumed known a priori. Besides, the measurements of inputs and outputs from a real structural system tend to be complex and expensive. Therefore, there is a significant interest in the development of an algorithm that uses as few measurements as possible to obtain the physical characteristics of the system without a priori knowledge of this system.

In this study, a parameter estimation technique based on SCE is presented to overcome some of the difficulties encountered in the field, which could be formulated as

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 107–113, 2009.

multimodal numerical optimization problems with high dimension. Four numerical examples are presented from which the effectiveness and efficiency of the SCE are investigated. The influence of incomplete availability of measurements on the performance of SCE for system identification is also discussed.

## 2 Problem Formulation

The basic idea in system identification is to compare the time dependent response of the system and a parameterized model by a norm or some performance criterion giving a measure to how well the model response fits the system response. Hence, the objective is to find a set of parameters that minimize the prediction error between system output *y(t)* , i.e., the measured data, and model output $\hat{y}(\theta,t)$ at each time-step *t*.

Therefore, our interest lies in minimizing the predefined error norm of the outputs, e.g., the following mean square error (MSE) function.

$$F(\theta)=\frac{1}{T}\sum_{k=1}^{T}\left\|y(k)-\hat{y}(k)\right\|^{2} . \tag{1}$$

where $\|\ \|$ represents the Euclidean norm of vectors. Formally, the optimization problem requires finding a vector $\theta \in R_n$, so that a certain quality criterion is satisfied, namely that the error norm $F(\theta)$ is minimized. The function $F(\theta)$ is commonly called a fitness function or objective function. In SCE, typically an objective function is used which reflects the goodness of solution. The identification problem thus is treated as a linearly constrained multi-dimensional optimization problem, namely

$$\begin{gathered}\min F(\theta), \theta=(\theta_1,\theta_2,...\theta_n)^T . \\ s.t. \theta \in S,\ S=\{\theta_{\min i} \le \theta_i \le \theta_{\max i};\ \forall i=1,2,...,n\} .\end{gathered} \tag{2}$$

where $F(\theta)$=objective function which maps decision variable $\theta$ into the objective space $F = R_n \to R$ , $S$ is the *n*-dimensional feasible search space, $\theta_{\max}$ and $\theta_{\min}$ denote the upper bounds and the lower bounds of the *n* parameters respectively.

## 3 Shuffled Complex Evolution Algorithm

In SCE, a population of NP (population size) solution vectors is initialized randomly at the start, which is evolved to find optimal solutions through the reflection, hongmutation, contraction, shuffling and selecting operation procedures. An optimization task consisting of *n* parameters can be represented by an *n*-dimensional vector. Let $S \in R_n$ be the search space of the problem under consideration. Then, the SCE algorithm utilizes NP, n-dimensional vectors

$$x_i=(x_{i1},x_{i2},...,x_{in})^T \in \theta,\ i=1,2,...,s . \tag{3}$$

as a population for each iteration, called a generation of the algorithm[4].

### 3.1 Implementation of SCE

The philosophy behind the SCE approach is to treat the global search as a process of natural evolution. The sampled points (s in number) constitute a population. The population is partitioned into several communities (complexes), each of which is permitted to evolve independently (i.e., search the space in different directions). After a certain number of generations, the communities are forced to mix, and new communities are formed through a process of shuffling. This procedure enhances survivability by a sharing of the information (about the search space) gained independently by each community.

### 3.2 CCE Strategy

Competitive complex evolution (CCE) strategy is the critical part of SCE algorithm. In CCE each member of a community (complex) is a potential parent with the ability to participate in a process of reproduction. A subcomplex selected from the complex is like a pair of parents, except that a subcomplex may consist of more than two members.

To ensure that the evolution process is competitive, the probability that better parents contribute to the generation of offspring is higher than that of worse parents. The use of a triangular probability distribution ensures this competitiveness.

Nelder and Mead's [8] procedure is applied to each subcomplex to generate most of the offspring. CCE uses the information contained in the subcomplex to direct the evolution in an improvement direction.

In addition, offspring are introduced at random locations of the feasible space under certain conditions in order to ensure that the process of evolution does not get trapped by unpromising regions. This is somewhat analogous to mutation in response to stress that can occur in biological evolution. Each mutation also helps to increase the amount of information stored in the sample.

Finally, each new offspring replaces the worst point of the current subcomplex, rather than the worst point of the entire population. This ensures that every parent gets at least one chance to contribute to the reproduction process before being replaced or discarded. Thus, none of the information contained in the sample is ignored.

### 3.3 Operational Parameters

According to the study result conducted by Duan [7], the SCE2 procedure is on the average the best algorithm. In most problems, it performs better than the SCE1, the other version of SCE. Thus, following Duan's suggestion, the several key parameters are set the same as the SCE2, where $m=(2n+1)$□$q=n+1$□$x=1$□$y=2n+1$. $m$ = the number of points in each complex, $q$=the number of points in each subcomplex, $x$=the number of generations that each subcomplex competitive evolution will take and $y$=the number of calculation times of each complex's evolution. As the result, $p$, which is the number of complexes, becomes the only parameter that requires user specified according to the difficulty of certain problem. When $p$ is relatively large, SCE is able to solve higher dimensional parameter estimation problem; however, at the same time, the method might become computationally demanding as well. Therefore, in this paper $p$ is set to 4.

## 4 Illustrative Examples

In this paper the problem is made to the most difficult case of unknown mass systems, where the mass, stiffness and damping of the structure are to be identified.

In order to assess the effectiveness of the parameter estimation technique with the SCE presented above, numerical simulations of five degrees of freedom (DOFs) structure system are carried out. The properties of the structural systems are given in Table 1. It is assumed that the system is excited by known forces and that the response of the structure, in terms of accelerations, is recorded at some given points.

**Table 1.** 5 DOF System Parameters

| *Stiffness (kN/m)* | |
|---|---|
| Level 1 | 2.485e5 |
| Level 2 | 1.921e5 |
| Level 3-5 | 1.522e5 |

| *Mass (kg)* | |
|---|---|
| Level 1-2 | 2.762e3 |
| Level 3-5 | 2.300e3 |

| *Damping (kN.s/m)* | |
|---|---|
| Level 1 | 3.129e3 |
| Level 2 | 2.536e3 |
| Level 3-5 | 2.030e3 |

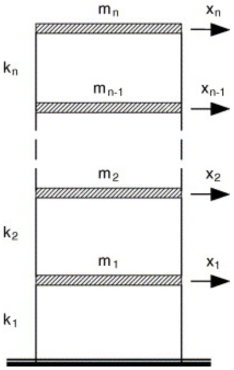


**Fig. 1.** n-DOF structural system

The dynamic equation of motion of a structural system can be written as

$$M\ddot{x} + C\dot{x} + Kx = u(t)\ . \tag{4}$$

where M, C and K are the mass ($m_1,\ldots,m_5$), damping($c_1,\ldots,c_5$) and stiffness ($k_1,\ldots,k_5$) matrices, x is the displacement vector and u is the input force vector.

Therefore, the system is fully described by the set of parameters

$$\theta = (m_1, m_2, \ldots, m_5;\ k_1, k_2, \ldots, k_5;\ c_1, c_2, \ldots, c_5)\ . \tag{5}$$

The influence of limited availability of measurements on the performance of SCE for parameters estimation is discussed. In the "full output" scenario, measurements at all DOFs are available, whereas in the second "partial output" scenario, only DOFs 2, 4, and 5 are available. The following cases of data availability will be treated here as:

1: Full set of the accelerations (outputs) available

$$y(t) = (\ddot{x}_1(t),\ \ddot{x}_2(t),\ \ldots,\ \ddot{x}_5(t))\ . \tag{6}$$

2: Partial set of the outputs available

$$y(t) = (\ddot{x}_2(t),\ \ddot{x}_4(t),\ \ddot{x}_5(t))\ . \tag{7}$$

The statistical simulation results of 5 independent runs for the known mass system with the usage of the SCE strategy and DE algorithm are carried out. The input and output (I/O) data are polluted (in the cases considering noise) with Gaussian, zero mean, white noise sequences, whose root mean-square (RMS) value is adjusted to be a certain percentage of the unpolluted time histories. The mean results of the parametric identification for full output scenario are summarized with 0% and 10% RMS noises.

In these scenarios, the mass distribution of the structure is supposed to be unknown priori. The time records used span a total length of 20 s with a sample time of 0.02 s. The SCE strategy parameter is $p$=4 and maximum number of function evaluation=20000. The search space is taken as 0.5–2.0 times the exact values.

All identified average results' errors are presented in Table 2. In addition, a typical SCE search performance for the 10% noise with partial measurements scenario is provided in Fig. 2, and a typical convergence characteristic of estimation for 4 scenarios is shown in Fig. 3. The following four scenarios tested includes:

CASE1: 0% noise + Full Outputs
CASE2: 0% noise + Partial Outputs
CASE3: 10% noise + Full Outputs
CASE4: 10% noise + Partial Outputs

**Table 2.** Results' errors for 4 cases

| Par | | CASE1 | | CASE2 | | CASE3 | | CASE4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | SCE | DE | SCE | DE | SCE | DE | SCE | DE |
| $m_1$~$m_5$ | Min err | 0.01 | 0.00 | 0.02 | 0.02 | 0.04 | 0.07 | 0.01 | 0.06 |
| | Max err | 0.03 | 0.09 | 0.21 | 0.16 | 0.37 | 0.31 | 0.29 | 0.52 |
| | Mean err | 0.02 | 0.04 | 0.11 | 0.07 | 0.27 | 0.18 | 0.14 | 0.29 |
| $k_1$~$k_5$ | Min err | 0.00 | 0.01 | 0.00 | 0.06 | 0.23 | 0.04 | 0.05 | 0.13 |
| | Max err | 0.04 | 0.09 | 0.12 | 0.49 | 0.42 | 0.29 | 0.70 | 0.68 |
| | Mean err | 0.02 | 0.05 | 0.06 | 0.29 | 0.30 | 0.14 | 0.34 | 0.36 |
| $c_1$~$c_5$ | Min err | 0.42 | 0.11 | 1.58 | 0.73 | 0.24 | 0.03 | 0.96 | 2.13 |
| | Max err | 0.12 | 0.97 | 4.66 | 2.56 | 2.74 | 2.58 | 6.52 | 7.82 |
| | Mean err | 0.26 | 0.70 | 3.00 | 1.75 | 1.01 | 1.28 | 3.82 | 3.45 |

Note: Relative errors of identification are in parentheses expressed in %.

From Table 2, without noise disturbing, it can be seen that the results obtained by both SCE and the DE are very close to the true values. The average results and errors obtained by DE a little outperform those obtained by the SCE. With 10% noise polluted, the values of estimated parameters obtained by SCE and DE are still very close to the true values of original parameters. These two schemes both seem to be powerful in escaping local optima and in search for the global optimum on complex problems.

From Figs. 2-3, we observe that for the cases without noise, in the first 50 generations, the convergences of estimation by SCE are faster than DE, but in the second 50 generations the DE outperforms the SCE. In the cases with 10% results, it is obvious that the SCE algorithms can get the true values much more quickly.

The results show that the SCE seems to perform well to the structural system identification problem, especially the polluted cases, yielding very accurate results but

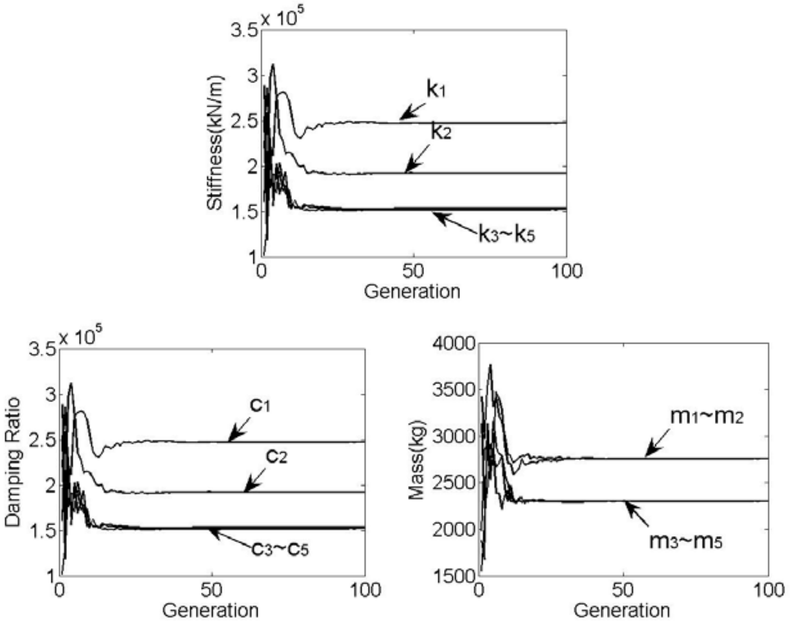
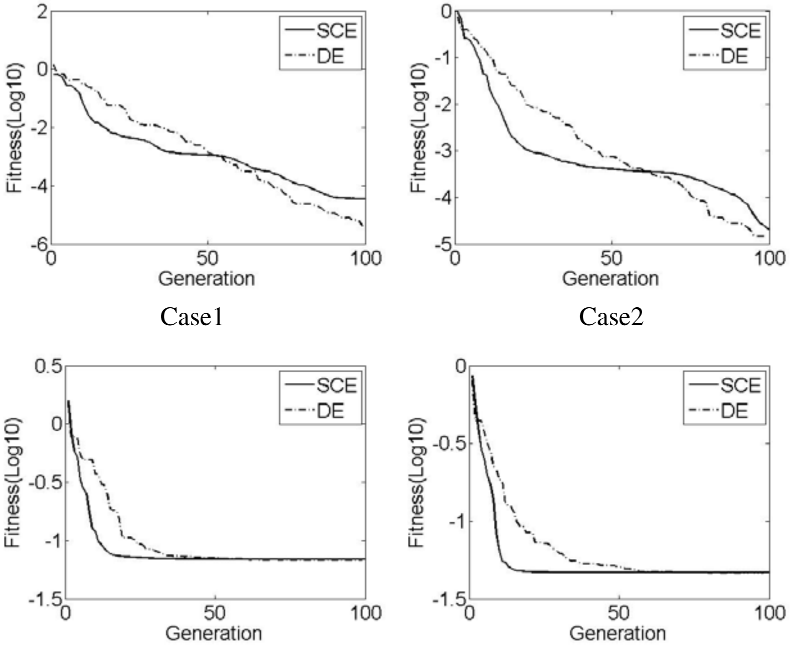


**Fig. 2.** Typical identification results for 5-DOF unknown mass system (partial measurement with 10% noise) by SCE



Case1

Case2

Case3

Case4

**Fig. 3.** One typical convergence characteristic of estimation for 4 scenarios

requiring less computing effort. The largest relative errors are usually observed in the damping coefficients. Due to the fact that the damping parameter has only a small contribution to the overall response, its value is generally poorly estimated. This is a fact that has been reported in other studies [1][2] as well.

In addition, the SCE approach is capable of locating the global optimum in all runs with in rather good results errors. It is well known that unknown mass systems are highly multimodal problems. Nevertheless, the maximum error of SCE is very good in damping of only 6.52% with partial output under 10% noise polluted in which DE reaches to 7.82%.

## 5 Conclusion

This paper has presented a shuffled complex evolution (SCE) strategy for the problem of structural system identification. SCE is very easy to implement and requires only one parameter tuning. Four numeric experiments have been conducted to assess the applicability of the SCE for structural parameters estimation. The results from our study show that SCE clearly and consistently performs excellent for hard unknown mass problems, both in respect to precision as well as robustness of the results. This proposed approach has no special requirements regarding the incomplete output measurements from the system. Even when the mass, stiffness and damping of the structure are unknown, the SCE can still converge to accurate results. The SCE approach is a promising tool for parameters estimation of structural systems in the sense that it is an optimal method requiring no prior knowledge on the structure.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China No. 50708076.

## References

1. Franco, G., Betti, R., Lus, H.: Identification of Structural Systems Using an Evolutionary Strategy. J. Eng. Mech. 130, 1125–1139 (2004)
2. Perry, M.J., Koh, C.G., Choo, Y.S.: Modified Genetic Algorithm Strategy for Structural Identification. J. Comput. & Struct. 84, 529–540 (2006)
3. Tolson, B.A.: Automatic Calibration, Management and Uncertainty Analysis: Phosphorus Transport in the Cannonsville Watershed. Cornell Univ., Ithaca (2005)
4. Duan, Q.: Global Optimization for Watershed Model Calibration. Mol. Water Sci. Appl. 6, 89–104 (2003)
5. Duan, Q., Gupta, V.K., Sorooshian, S.: Effective and Efficient Global Optimization for Conceptual Rainfall Runoff Models. Water Resour. Research 28, 1015–1031 (1992)
6. Duan, Q., Gupta, V.K., Sorooshian, S.: Shuffled Complex Evolution Approach for Effective and Efficient Global Minimization. J. Optim. Theory Appl. 76, 501–521 (1993)
7. Duan, Q., Sorooshian, S., Gupta, V.K.: Optimal Use of the SCE-UA Global Optimization Method for Calibrating Watershed Models. J. Hydrol. 158, 265–284 (1994)
8. Nelder, J.A., Mead, R.: A Simplex Method for Function Minimization. Comput. J. 7, 308–313 (1965)

# A Novel Evolving Clustering Algorithm with Polynomial Regression for Chaotic Time-Series Prediction

Harya Widiputra[1], Henry Kho[2], Lukas[3], Russel Pears[1], and Nikola Kasabov[1]

[1] Knowledge Engineering and Discovery Research Institute,
Auckland University of Technology, New Zealand
{harya.widiputra,russel.pears,nikola.kasabov}@aut.ac.nz
[2] Faculty of Information and Technology, Swiss German University,
BSD City, Indonesia
{henry}@student.sgu.ac.id
[3] Cognitive Engineering Research Group (CERG), Universitas Katolik Atma Jaya,
Indonesia
{lukas}@atmajaya.ac.id

**Abstract.** Time-series prediction has been a very well researched topic in recent studies. Some popular approaches to this problem are the traditional statistical methods e.g. multiple linear regression and moving average, and neural network with the Multi Layer Perceptron which has shown its supremacy in time-series prediction. In this study, we used a different approach based on evolving clustering algorithm with polynomial regressions to find repeating local patterns in a time-series data. To illustrate chaotic time-series data we have taken into account the use of stock price data from Indonesian stock exchange market and currency exchange rate data. In addition, we have also conducted a benchmark test using the Mackey Glass data set. Results showed that the algorithm offers a considerably high accuracy in time-series prediction and could also reveal repeating patterns of movement from the past.

**Keywords:** evolving clustering algorithm, polynomial regression, chaotic time-series data.

## 1 Introduction

Chaotic time-series prediction has become a very well-liked topic for research and there have been a number of methods from different fields introduced before to solve this problem. Some very popular approaches for time-series prediction come from traditional statistic approaches, i.e. linear regression, multiple linear regressions, and moving average [5]. Other than that, various models and algorithms from the machine learning and data mining fields have also been introduced (e.g. Multi Layer Perceptron, Pattern Recognition, and Support Vector Machine) [4], [8], [9]. All of these approaches have shown their strength in solving the time-series prediction problem by presenting a good-quality of accuracy in predicting future values, however not much new knowledge can be revealed and learned from these models.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 114–121, 2009.

A breakthrough was made by [7] with the Dynamic Evolving Neuro-Fuzzy Inference System (DENFIS) in which the algorithm was not only able to give good accuracy in predicting movement of time-series data, but it was also able to extract new knowledge in the form of sets of fuzzy rules which govern the movement of the series. This has stimulated us to make an effort to mine knowledge about the existence of repeating patterns in a time-series movement, particularly in a chaotic time-series data (e.g. stock prices, exchange rates), and then to use extracted knowledge to predict future movement of the series. We are aiming to represent the rules in the form of repeating polynomial functions, group them based on their likeness, and do prediction for future value by trying to discover similar patterns from the past that can be associated with current condition. In addition, we also expect the algorithm to acquire the ability to evolve when new data sample comes, therefore it would be able to adapt when new problems emerge.

In this study, we used in general two different types of data set as experimental data, which are; (1) the financial data set consists of the Indonesian stock prices data and some currency exchange rates as these data can be considered as a Complex Dynamic System (CDS) [5], and (2) a benchmark data set which in this case is the Mackey Glass data set.

## 2 Clustering as Local Modeling

One of the challenges in information science is to be able to represent dynamic systems, model them and then to reveal the rules that govern the behavior of the variables over time [3]. An option to build a complete model of a dynamic system is by clustering comparable problem sets from a complete problem space based on certain condition into different groups. A model then can be created for each group (i.e. a local regression), and by combining all models from each cluster it will constitute a complete model which might be able to cover all problems from the complete problem space [1], [2], [6]. Similarity in this case is usually (but not limited to) determined by calculating a Euclidean distance between two data samples [5].

We developed a new algorithm which will extract pattern from a localized data set or a chunk of data (as a sub-space problem from a complete problem space) using a polynomial regression. Extracted patterns (in forms of polynomial functions) then will be grouped based on the likeness of their shape (trend of movement). We believe that repeating patterns of movement exist in a time-series data based on [2], [5], [7], even in a chaotic one, and it can be used to predict future values.

### 2.1 Similarity and Distance Measurement

Defining similarity between two objects in clustering process is a very important step. Different definition about similarity would give us diverse solutions or various groups of data. In a clustering process, it is common to use the Euclidean distance to measure similarity between objects [5]. This can be done by calculating distance from each attributes that we used to describe the object.

Nonetheless, in our algorithm it will not be suitable to apply the Euclidean distance, as what we would like to group are not objects but sets of polynomial function which represent patterns of movement of a time-series data across time. Therefore, we applied a different method to calculate similarity between two polynomial functions, which is the Angular Separation or Cosine distance (equation 1). By using the Cosine distance, we were able to measure similarity between two polynomial functions based on the shape and direction of the function. As for the attributes that we used to describe a polynomial function, we took into consideration the coefficients of each part of the polynomial function.

$$S_{ij} = \frac{\sum_{k=1}^{n} x_{ik}.x_{jk}}{\left(\sum_{k=1}^{n} x_{ik}^{2} . \sum_{r=1}^{n} x_{jr}^{2}\right)^{\frac{1}{2}}} \tag{1}$$

### 2.2 ECM as Evolving Clustering Algorithm

Evolving Clustering Method (ECM) was introduced by [6]. ECM is a fast, one-pass algorithm for dynamic clustering of an input stream of data. It is a distance-based clustering method where the cluster centres are represented by evolved nodes in an on-line mode. In the clustering process, the data samples come from a data stream and this process starts with an empty set of clusters. As new data samples come, new clusters are created or if the new data sample is similar to those previous ones, it will be added into one of the existing clusters and the cluster centre will be updated.

ECM is used in Dynamic Evolving Neuro-Fuzzy Inference Systems (DENFIS) [7]. In our proposed algorithm, we have also implemented ECM as the core evolving clustering algorithm. Nevertheless, we have also made a fundamental change in the algorithm. ECM uses the Euclidean distance to measure similarity between objects when it creates or updates clusters. In our algorithm, we replaced the Euclidean distance with Cosine distance. The main reason of this modification is explained in sub-chapter 2.1.

### 2.3 DyCPR as a Novel Evolving Clustering Algorithm for Time-Series Prediction

We named the proposed algorithm Dynamic Clustering with Polynomial Regression (DyCPR). The main idea behind the algorithm is to extract patterns of movement from a time-series data using a polynomial regression. After the patterns have been extracted, it will be grouped using the modified ECM algorithm. Cluster centres will represent the average movement calculated as a superposition function from all polynomial functions belong to the cluster. These cluster centres will be used as references to predict movement of the series by finding a centre that can be related with current condition. Complete algorithm of DyCPR is described as follow;

- **Step 0:** First step of the algorithm is considered to be a data pre-processing step. In this step the algorithm, we will calculate the difference of values from the time- series data. $\boldsymbol{diff_t = x_{t+1}\text{-}x_t}$, where $\boldsymbol{diff_t}$ is value difference at time $\boldsymbol{t}$ and $\boldsymbol{X}$ are actual value at time point $\boldsymbol{t+1}$ and $\boldsymbol{t}$ respectively. The motive why the algorithm calculates values difference from time-series data is, because it is the movement of values difference that will be predicted instead of the actual value.

- **Step 1:** After values difference has been calculated, the next step of the algorithm is to find the best-fit regression function from a chunk of data, $X_i$ with size $\boldsymbol{n}$. For every chunk of data which comes in, the best-fit regression function will be extracted by calculating linear regression and polynomial regression up to certain order (i.e. up to 6th order). $PF_i = \{LR_i, PR1_i, PR2_i, PR3_i, PR4_i, PR5_i, PR6_i\}$ best-fit regression function is defined by calculating $bestPF = minMSE(PF_i)$. Besides calculating the best-fit regression function from the chunk of data with size $\boldsymbol{n}$, the algorithm will also calculated best-fit regression function for next movement, $PFM_i$ using another chunk of data with the size $\boldsymbol{n}$**+1**, $Y_i$.
- **Step 2:** For the first chunk of data, found ***bestPF*** will become the cluster centre of the first cluster, $Cc_j = bestPF_i$ and the function to predict next movement of data samples that belong to that cluster is set to $PFM_i$. The radius of the cluster, $Ru_j$ is set to 0. If there is no more chunks of data comes then the algorithm stops. If there are still more chunks of data then the algorithm returns to **Step 1**, where the best-fit regression function will be calculated again. After the $bestPF_i$ is calculated for $X_i$, similarity between $bestPF_i$ with the ***Cc*** is measured. Similarity is calculated using Cosine distance by taking coefficients of $bestPF_i$ function and ***Cc*** function into account.
- **Step 3:** The forth step of the algorithm is basically the implementation of the ECM algorithm, where the algorithm will calculate distance between $bestPF_i$ to all existing cluster centres, ***Cc*** and makes update to cluster centre ***j***, $Cc_j$ when it is found that distance between $bestPF_i$ and $Cc_j$ is less than **2x*Dthr***. The process of updating the value of cluster radius, $Ru_j$ is the same as it is in ECM. However DyCPR will update also the best-fit regression function for next movement of cluster ***j***, $PFM_j$ by calculating superposition value between existing $PFM_j$ and $PFM_i$ when it decides that $bestPF_i$ belongs to cluster ***j***. If all chunks of data sample have been processed than the algorithm stops, else it returns to **Step 1**.
- **Step 4:** In the prediction step, the algorithm will calculate the best-fit regression function from current chunk of data with size $\boldsymbol{n}$, $X_t = \{x_t, x_{t-1}, x_{t-2}, \ldots, x_{t-(n-1)}\}$. Calculated best-fit regression function for time ***t***, $bestPF_t$, then will be compared to all existing cluster centres, ***Cc***. When closest $Cc_j$ is found, then $PFM_j$ will be used as a reference function to calculate next movement of values differences, $px_{t+1}$. Predicted actual value then will be calculated as, $x_{t+1} = x_t + px_{t+1}$. Learning process then will be continued by taking new data sample as a data sample to update existing clusters and the algorithm returns to **Step 1**.

One key characteristics of DyCPR (other than its power to evolve by creating and updating clusters when new data sample becomes available) is that the algorithm will do prediction of future values by predicting the polynomial function that will represent movement of the series in the future (this is explained in Step 1 and 3 in the DyCPR algorithm). This is significantly different in comparison to DENFIS where the algorithm does prediction by calculating real predicted values based on extracted fuzzy rules represent by each cluster centres [7].

## 3 Experiments and Results

In this study, we used data from financial field as experimental data. These data are; (1) data of five stock prices from Indonesian stock market (namely TLKMJK, AALIJK, LMASJK, KLBFJK, and ELTYJK; data are available from: http://finance.yahoo.com/q/cp?s=^JKSE) and (2) currency exchange rates data (i.e. NZD to USD and IDR to USD)[1]. The five stock prices data represent three different types of movement, which are; chaotic, increasing and decreasing. The stock prices data cover period of January 2005 to January 2009, while the currency exchange rates data expand from January 2006 to January 2009. As for benchmark data, we use the Mackey Glass data set to examine the robustness of the algorithm. Figure 1 illustrates movement of the Indonesian stock price data for ELTYJK data.

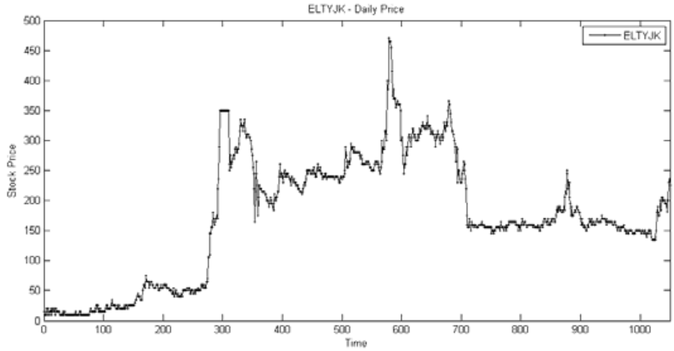


**Fig. 1.** Daily basis data (closing price) of ELTYJK in Indonesian stock exchange market spanning from January 2005 to January 2009

No normalization process applied to the data before the training process is started. On the other hand, what will be extracted are patterns of movement of difference between values across time instead of movement of the actual values (as it has been explained in Step 1 in DyCPR algorithm).

Results showed that the algorithm demonstrated its strength by giving a considerable high accuracy of prediction results (with RMSE of 2.4680 and 0.2449 for ELTYJK data and LMASJK data respectively; please note the RMSE is un- normalized RMSE). This can be seen in figure 2 and 3, where we attached prediction results for two stock prices data with different behavior which are; (1) ELTYJK data with increasing trend and (2) LMASJK data with decreasing trend, here we used 80% of total data set records as training set and 20% as test set. Even more, the algorithm showed its ability to adapt to new problems (unusual pattern of movement) which did not appear during the training step as in the case of LMASJK data for the last view points (i.e. a flat-line pattern).

[1] NZD is New Zealand Dollar, USD is United States Dollar and IDR is Indonesian Rupiah.

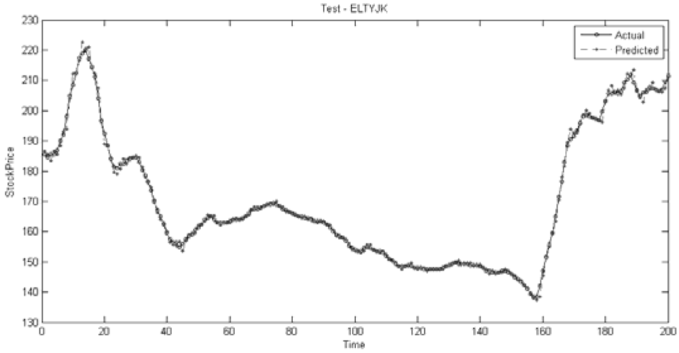


**Fig. 2.** Comparation of predicted value (in dashed line) and actual value (in solid line) for ELTYJK data set on testing set. Number of training set is 850 and number of testing set is 200. To predict value at *t*+1, values from time point *t*, *t*-1, *t*-2, *t*-3, *t*-4, *t*-5, *t*-6, *t*-7, *t*-8, and *t*-9 are used to find the best fit polynomial regression. Distance threshold used in DyCPR is 0.3.

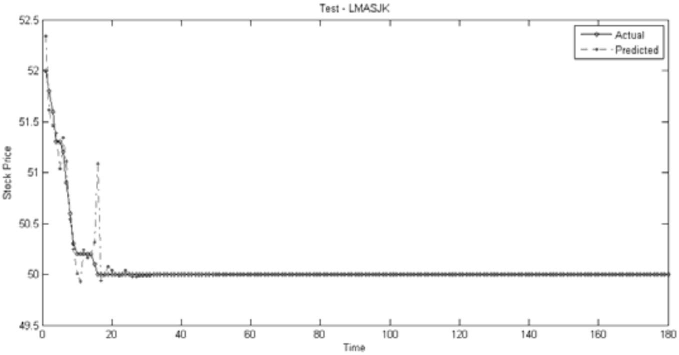


**Fig. 3.** Comparation of predicted value (in dashed line) and actual value (in solid line) for LMASJK data set on testing set. Number of training set is 750 and number of testing set is 180. To predict value at *t*+1, values from time point *t*, *t*-1, *t*-2, *t*-3, *t*-4, *t*-5, *t*-6, *t*-7, *t*-8, and *t*-9 are used to find the best fit polynomial regression. Distance threshold used in DyCPR is 0.3.

As it is expected, the algorithm was not only able to provide excellent accuracy in predicting future values, it was also capable to extract new knowledge in the form of repeating polynomial functions which exist in the series over time. This can be seen in the clusters the algorithm created as showed in figure 4 for the ELTYJK data.

From figure 4, we can observe how the polynomial functions are being grouped into different clusters, and how the cluster centres represent the average movement function for each group. This finding justified our assumption that repeating patterns of movement do exist in a time-series data, and that it can be modeled using a polynomial functions. We have also conducted a benchmark test using the Mackey Glass data set as part of our experiment. Complete results of our experiments with the other data sets can be seen in table 1.

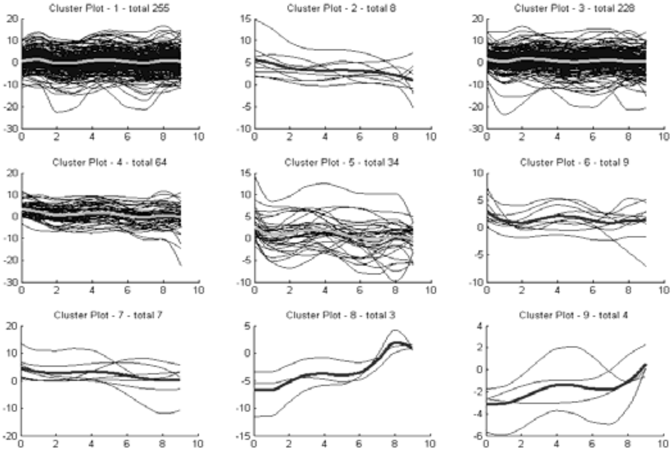


**Fig. 4.** Created clusters during training and testing process when predicting movement of stock price difference for ELTYJK data. Each clusters show comparable patterns of movement based on the likeness of shape and direction of extracted polynomial regression for each chunk of data. Cluster centres (in wider line) represent movement of all polynomial functions which belong to a cluster, are calculated as superposition functions.

Results from table 1 showed that the algorithm demonstrated its superiority in chaotic time-series prediction. The algorithm is also effective to predict movement of different types of time-series data with different behaviours (e.g. fluctuative, increasing, and decreasing).

**Table 1.** Results of DyCPR used to predict 8 different types of data set covering stock prices data set, currency exchange rate data set, and the Mackey Glass data set. 80% of data is used as training set and 20% as testing set for each experiments. Value of distance threshold, *Dthr* in DyCPR is set to 0.3.

| Data Set | Category | Trend | Range of Values | RMSE |
|---|---|---|---|---|
| TLKMJK | Indonesian Stock Price | Increasing | 3,853.9-11,933.0 | 34.0077 |
| AALIJK | Indonesian Stock Price | Fluctuate | 346.0-32,026.0 | 109.3862 |
| LMASJK | Indonesian Stock Price | Decreasing | 35.0-1,304.8 | 0.2449 |
| ELTYJK | Indonesian Stock Price | Fluctuate | 331.75-1,508.4 | 2.4680 |
| KLBFJK | Indonesian Stock Price | Fluctuate | 10.0-680.0 | 3.1927 |
| NZDUSD | Exchange Rate | Balanced | 1.2235-1.9058 | 0.0083 |
| IDRUSD | Exchange Rate | Decreasing | 8,684.0-12,209.0 | 72.6284 |
| Mackey Glass | Benchmark Data | Fluctuate | 0.2192-1.3137 | 0.0057 |

## 4 Conclusion and Future Works

Dynamic Clustering with Polynomial Regression (DyCPR) showed its strength by giving high accuracy in predicting movement of chaotic time-series data. The robustness of the algorithm has also been tested by conducting experiment using different data sets with different behaviours. Furthermore, proposed algorithm was also able to extract repeating patterns of movement from chaotic time-series data. It has also been verified that by using polynomial regressions to model patterns of movement in a localized sub-space of time-series data, we were able to acquire prediction of next movement with a considerably high accuracy.

As for future works, we would like to extend the algorithm so it can be used to predict movement of not only a single chaotic time-series data but for multiple time-series data as well. One possibility to achieve this is by implementing a recursive clustering method, where the first clustering process will group data from different series into a number of groups with comparable behaviour, and the second clustering process will be to extract patterns of movement from created clusters in the first step.

## References

1. Kasabov, N.: Adaptation and interaction in dynamical systems: Modelling and rule discovery through evolving connectionist systems. Applied Soft Computing 6, 307–322 (2006)
2. Kasabov, N.: Global, local and personalised modelling and pattern discovery in bioinformatics: An integrated approach. Pattern Recognition Letters 28, 673–685 (2007)
3. Kasabov, N.: Evolving Connectionist Systems: The Knowledge Engineering Approach. Springer, Heidelberg (2007)
4. Kim, T., Adali, T.: Approximation by Fully Complex Multilayer Perceptrons. Neural Computation 15, 1641–1666 (2003)
5. Serguieva, A., Kalganova, T., Khan, T.: An intelligent system for risk classification of stock investment projects. Journal of Applied Systems Studies 4(2), 236–261 (2003)
6. Song, Q., Kasabov, N.: ECM – A Novel On-line Evolving Clustering Method and Its Applications. In: Posner, M.I. (ed.) Foundations of cognitive science, pp. 631–682 (2001)
7. Song, Q., Kasabov, N.: Dynamic evolving neuro-fuzzy inference system (DENFIS): On-line learning and application for time-series prediction. IEEE Transactions of Fuzzy Systems 10, 144–154 (2002)
8. Yang, H., Chan, L., King, I.: Support Vector Machine Regression for Volatile Stock Market Prediction. In: Yellin, D.M. (ed.) Attribute Grammar Inversion and Source-to-source Translation. LNCS, vol. 302, pp. 143–152. Springer, Heidelberg (1988)
9. Zanghui, Z., Yau, H., Fu, A.M.N.: A new stock price prediction method based on pattern classification. In: International Joint Conference on Neural Network 1999, pp. 3866–3870 (1999)

# A Multi-strategy Differential Evolution Algorithm for Financial Prediction with Single Multiplicative Neuron

Chukiat Worasucheep[1] and Prabhas Chongstitvatana[2]

[1] Applied Computer Science, Department of Mathematics, Faculty of Science,
King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand
chukiat.wor@kmutt.ac.th
[2] Department of Computer Engineering, Faculty of Engineering,
Chulalongkorn University, Bangkok 10330, Thailand
prabhas.c@chula.ac.th

**Abstract.** This paper proposes a differential evolution (DE) algorithm that combines the strengths of multiple strategies together. The selection of strategy and control parameters for each individual happens every learning period. Thus the user gains the benefits of different strategies without difficult fine tuning of control parameters. The performance of the proposed MDE algorithm is evaluated on well-known benchmark functions and is superior to some other efficient and widely used variants of DE. In addition, MDE is applied to optimize both weights and biases of a single multiplicative neuron for prediction of DJIA with 3228 samples. Experiments show its better performance than other methods in learning ability and generalization.

**Keywords:** Differential Evolution, Evolutionary Algorithm, Single Multiplicative Neuron, Financial Prediction.

## 1 Introduction

Differential Evolution (DE) algorithm is an efficient evolutionary algorithm (EA), proposed by Storn and Price, for global optimization in continuous space [1]. DE has been widely applied and shown its strengths in many application areas from scientific, engineering, to financial [2-3]. DE has three critical control parameters: crossover rate (CR), mutation factor (F), and population size (NP), that require a proper setting to ensure its performance and convergence. Different proper ranges of CR are suggested for different optimization functions. Mutation factor F usually affects the convergence speed. NP is normally allowed the user to set to handle the complexity or dimensionality of the function. In addition, DE has a number of perturbation methods or mutation strategies developed. Most of them are suitable for different characteristics of functions. While some strategies have shown good convergence for unimodal functions, others are advised for complex multimodal functions. This complex task of the strategy selection and parameter settings becomes burdens to many practitioners, and thus encourages a number of research groups to attempt to automate the chore.

This paper proposes a simple method of mixing different strategies of DE to pull out their benefits. Trigonometric mutation operator, by Fan and Lampinen [4], and

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 122–130, 2009.

two other widely used strategies are first observed. The three strategies and parameter CR are self-adapted during the learning period. The proposed multi-strategy DE (or MDE) is evaluated using 5 nonlinear benchmark functions widely used in optimization literature. Then the algorithm is applied to optimize weights and biases of a single multiplicative neuron (SMN) [5] for the prediction of next-day closing index of Dow Jones Industrial Average (DJIA) during Jan 1994 to October 2006. The dataset are technical indicators computed from the five inputs: daily opening, high, low, closing indexes and the volume traded each day. The results are compared with those from back propagation and conventional DE variants.

This article is organized as follows. The next section briefly reviews basic understanding of the related topics, i.e., DE and SMN for financial prediction. Section 3 describes the proposed algorithm which is then evaluated with benchmark functions in Section 4. Section 5 describes the application in optimizing the SMN for the prediction of DJIA index. Finally Section 6 concludes this work.

## 2 DE and Time-Series Prediction with SMN

This section briefly reviews essential understanding of DE, its adaptation, and the financial prediction using SMN optimized with EAs.

### 2.1 Differential Evolution and Its Adaptation of Strategies and Parameters

Differential evolution (DE) is a population-based and directed search method [1]. Like other EAs, it starts with a randomly generated initial population of individuals, called vectors in DE domain. DE generates offspring by perturbing the solutions with a scaled difference of two randomly selected parent vectors. Then the replacement of an individual occurs only if the offspring outperforms its corresponding parent. This process will be iterated until some stopping criteria such as a maximum number of generations or number of objective function calls allowed [2]. There exist many mutation strategies in DE algorithms resulting in different variants, whose names are denoted with *DE/x/y/z* naming scheme [1-2]. The performance of the conventional DE algorithm highly depends on the chosen mutation strategy and associated parameter values (NP, CR, and F). In the past decade, DE researchers have suggested many empirical guidelines for choosing mutation strategies and tuning of the control parameter. Unfortunately various conflicting conclusions have been drawn with regard to the rules for manual configuration. Therefore researchers have developed some techniques to avoid such manual tuning. For examples, Qin et al. [6] recently proposed a self-adaptation of five conventional mutation strategies as well as parameters CR and F during the learning period. A similar work by Brest et al. [7] also encoded F and CR into each vector but randomized and adjusted them with different ranges and methods.

### 2.2 The Single Multiplicative Neuron for Financial Prediction

Various neural network models and training algorithms have been used for time series prediction [8-9]. Since most financial time-series are nonlinear in nature, multiplication being the most basic unit of nonlinearities has been a natural choice of models in the artificial neuron model for financial prediction. In this work a single multiplicative neuron (SMN) [5] is employed because SMNs are easy to implement by using the standard back propagation (BP) learning algorithm and exhibit better performance than the multilayered neural networks with special structures [5,8]. The structure of a generalized SMN model with learning algorithm is briefly as follow. Let $(x_1, x_2, \ldots, x_n)$, $(w_1, w_2, \ldots, w_n)$ and $(b_1, b_2, \ldots, b_n)$ be the input pattern, weights and biases of the model, respectively. The output function is the *logsig* of the operator Ω which is a multiplicative operation as in Eq. (1) and $u$ is equal to Ω.

$$y = \frac{1}{1+e^{-u}}, \text{ where } u = \Omega = \prod_{i=1}^{n} (w_i x_i + b_i). \tag{1}$$

Back propagation (BP) algorithm is generally adopted as learning algorithm to train the SMP. BP is used to minimize the error function defined in Eq. (2).

$$E = \frac{1}{2}\sum_{p=1}^{N} (y_p - d_p)^2, \tag{2}$$

where $d_p$ represent the desired network output for the $p$th input pattern, and $y_p$ is the computed output neuron. Using the steepest descent gradient approach and the chain rules for the partial derivatives, the update rules for the weights and biases of the model are given in Eqs (3) and (4), respectively.

$$w_i^{new} = w_i^{old} - \eta \frac{1}{N}\sum_{p=1}^{N}\left( (y_p - d_p) y_p (1 - y_p) \frac{u}{w_i x_i + b_i} x_i \right), \tag{3}$$

$$b_i^{new} = b_i^{old} - \eta \frac{1}{N}\sum_{p=1}^{N}\left( (y_p - d_p) y_p (1 - y_p) \frac{u}{w_i x_i + b_i} \right), \tag{4}$$

where $\eta$ is the learning rate parameter that controls the convergence speed. The above process is iterative until some termination criteria are met such as the maximum generations.

Yadav et al. [5] recently applied SMN to predict a set of well-known time-series including a currency exchange rate from 2002 to 2004 (totaling of 800 samples). The result has outperformed a multilayer neuron network. Soon after that, Zhao and Yang [9] proposed a Particle Swarm Optimization as learning algorithm of SMN for three well-known time series prediction problems. Their results outperformed those from learning with BP and Genetic Algorithm.

## 3 The Proposed Multi-strategy Differential Evolution

This section describes the proposed multi-strategy DE (MDE) that combines the strength of various DE strategies. As discussed in Sections 1 and 2.1, different mutation strategies are suggested depending on the characteristics or complexity of the function at hands, which in most cases are usually unknown a priori. In this work, we have chosen three common strategies as follows.

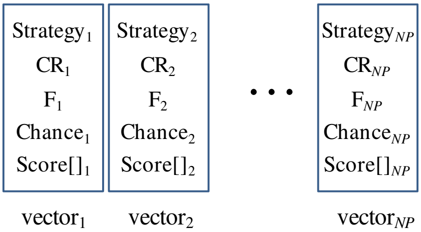
1. The DE/rand/1/bin (RAND) strategy which is the most widely used strategy and regarded as the standard DE. RAND is has good exploration ability and is suitable for multimodal functions [2-3].

2. The DE/current-to-best/1/bin (CURR2BEST) strategy has demonstrated a good convergence for a wide set of functions [2, 6].
3. The trigonometric mutation operator (TMO) performs as a rather greedy local search [4]. A small value of mutation probability parameter was suggested in [4] for a general use to balance the convergence speed (of TMO) and exploration ability (of RAND) in TDE.

This set of strategies is chosen based on an idea that an efficient mix of strategies should have diverse characteristics. However, other mixes of various strategies are under investigation and are expected to provide a better performance and robustness.

The structure of DE population in MDE is as illustrated in Fig. 1 that excludes the $D$-dimensional position $\{x_1, x_2, \ldots , x_D\}$ and the objective function value $f$ of each vector. For each vector $i$, Strategy$_i$ indicates one of the available strategies. The current number of available strategies in this work (*NS*) is 3. CR$_i$ is set for the corresponding strategy as widely suggested in DE literature i.e., 0.9 for RAND and CURR2BEST [1-3,6], and 0.95 for TMO [4]. Parameter F is related to convergence speed and most suggested to be randomized within a range of (0, 1+]. In this work, the value of F$_i$ is randomized from a uniform distribution within [0.2, 1.1] to cover both exploration and exploitation capability.

Chance$_i$ is a probability value determining the chance that vector $i$ chooses one of the mutation strategies. At the beginning of evolutionary process, Chance$_i$ is initialized equally to 1/*NS*, where the current *NS* is 3 in this work. The roulette wheel selection method is utilized to probabilistically select one of the available strategies based on Chance$_i$. For the next generations, all Chance$_i$ are recalculated every learning period $\lambda$ as follows. Each vector $i$ has an array Score[]$_i$ that continuously collects a flag indicating a success or improvement of the vector $i$'s objective function value $f_i$. That is, if at generation $g$, vector $k$'s $f_k$ has improved after the mutation and crossover operation, Score[$g$]$_k$ is set to 1, and 0 otherwise. This array Score[]$_i$ is thus a storage for collecting or counting the number of improvement – the $f$ of that vector $i$ has improved – over the learning period. The size of the array Score[]$_i$ is limited to the learning period $\lambda$. Windowing scheme is used for Score[]$_i$ to let the learning mechanism able to keep only the most recent $\lambda$-generation information.



| Strategy$_1$ | Strategy$_2$ | | Strategy$_{NP}$ |
|---|---|---|---|
| CR$_1$ | CR$_2$ | | CR$_{NP}$ |
| F$_1$ | F$_2$ | • • • | F$_{NP}$ |
| Chance$_1$ | Chance$_2$ | | Chance$_{NP}$ |
| Score[]$_1$ | Score[]$_2$ | | Score[]$_{NP}$ |
| vector$_1$ | vector$_2$ | | vector$_{NP}$ |

**Fig. 1.** The structure of the population in MDE

For any population-based search and optimization algorithms, maintaining diversity is of great importance. Some strategies to be included in MDE, such as TMO, can possess a high convergence. Therefore, the population initialization routine uniformly randomized creates vectors of 5 times of population size (5×NP). Then one of them is randomly selected as a real population vector. Each of the remaining vectors is gradually selected into the population, only if the summation of its Euclidean distances to all vectors already in the population is of minimum. This scheme will enhance diversity at the beginning.

# 4 Experimentation

This section describes the experimentation to evaluate the performance of the proposed MDE algorithm by using a set of four widely-used benchmark functions for minimization. Table 1 illustrates the descriptions of those functions whose global minima are 0. The first two functions are unimodal while the remaining two functions are multimodal. The experiment will test MDE algorithm using 50 dimensions or decision variables. The maximum numbers of function calls (MAXNFC) are limited to 10000D = 500,000. The basic statistical results from executing 30 independent runs with different seed numbers for random number generator are collected. The results will be compared to those from the standard DE/rand/1/bin (CR=0.9, F=0.5) and TDE (CR=0.95, F=0.99). The associated parameter values are set as widely suggested in literature [1-3]. In all cases, the experiment investigates the results for two different values of NP, i.e. 4D and 8D. The learning period λ of MDE is set to 50 after a preliminary test of the selected functions with varying values of λ from 30 to 70 with a step of 10.

## 4.1 Experimental Results

Table 2 shows the means, standard deviations, the best, and the worst results from 30 runs of each algorithm for each case. Fig. 2 illustrates averages convergence graphs for each case. From the table and the figure, we can observe the following results.

1. For the simple F1 Schwefel 2.22 function, MDE clearly outperforms both RAND and TDE with both NP values tested. This demonstrates a good convergence of MDE for functions with a simple landscape.

**Table 1.** Search Ranges of The Minimization Problems

| Function | | | Search Range |
|---|---|---|---|
| Schwefel 2.22 | F1(x) = | $\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$ | [-10, 10] |
| Rosenbrock's | F2(x) = | $\sum_{i=1}^{D-1}\left[100(x_{i+1}-x_i^{2})^2+(1-x_i)^2\right]$ | [-30, 30] |
| Rastrigin's | F3(x) = | $\sum_{i=1}^{D}\left[x_i^{2}-10\cos(2\pi x_i)+10\right]$ | [-5.12, 5.12] |
| Masters (Inverted cosine wave function) | F4(x) = | $-\sum_{i=1}^{D}\left(\exp\left(\frac{-(x_i^2+x_{i+1}^2+0.5x_i\,x_{i+1})}{8}\right)\right)$ <br> $\times\cos\left(4\sqrt{x_i^2+x_{i+1}^2+0.5x_i\,x_{i+1}}\right)+n-1$ | [-5, 5] |

**Table 2.** Statistical Results for 50 Dimensions

| Function | Algorithm | NP | Mean | S.D. | Best | Worst |
|---|---|---|---|---|---|---|
| F1 Schwefel 2.22 | RAND | 200 | 1.87E-04 | 5.64E-05 | 9.77E-05 | 3.20E-04 |
| | RAND | 400 | 3.717 | 1.030 | 2.466 | 6.586 |
| | MDE | **200** | **9.03E-19** | **3.92E-18** | **2.49E-26** | **1.80E-17** |
| | MDE | **400** | **3.99E-15** | **1.24E-14** | **8.88E-18** | **5.73E-14** |
| | TDE | 200 | 1.71E-06 | 4.42E-07 | 1.04E-06 | 2.85E-06 |
| | TDE | 400 | 0.119 | 0.031 | 0.071 | 0.176 |
| F2 Rosenbrock's | RAND | 200 | 34.316 | **0.797** | 32.994 | 36.467 |
| | RAND | 400 | 47.561 | **0.405** | 46.860 | 48.346 |
| | MDE | **200** | **1.046** | 1.677 | **1.67E-08** | **4.034** |
| | MDE | **400** | **11.913** | 4.090 | **3.359** | **18.948** |
| | TDE | 200 | 44.537 | 16.791 | 34.678 | 92.721 |
| | TDE | 400 | 45.718 | 1.465 | 43.179 | 49.186 |
| F3 Rastrigin's | RAND | 200 | 237.031 | 19.692 | 196.060 | 282.849 |
| | RAND | 400 | 342.739 | 12.363 | 319.394 | 369.747 |
| | MDE | **200** | **13.651** | **3.642** | 7.960 | **23.879** |
| | MDE | **400** | **9.711** | **3.022** | 4.974 | **16.914** |
| | TDE | 200 | 28.560 | 47.051 | **2.985** | 141.846 |
| | TDE | 400 | 31.739 | 55.178 | **2.024** | 207.077 |
| F4 Masters | RAND | 200 | 36.821 | **0.653** | 35.505 | 37.788 |
| | RAND | 400 | 37.342 | 1.062 | 35.103 | 38.447 |
| | MDE | **200** | **22.805** | 6.817 | **10.491** | **30.645** |
| | MDE | **400** | **30.410** | 3.399 | **20.515** | **32.878** |
| | TDE | 200 | 35.901 | 1.197 | 31.798 | 37.169 |
| | TDE | 400 | 36.132 | **0.871** | 34.287 | 36.739 |

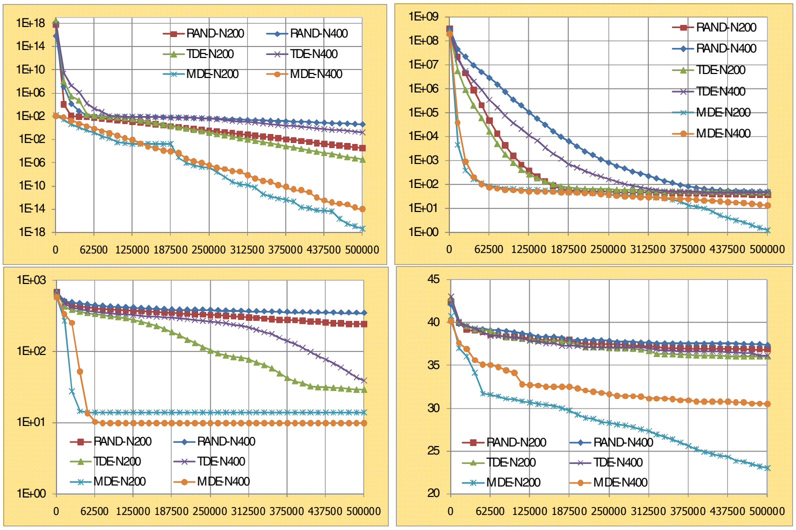


**Fig. 2.** Average convergence graphs: F1 Schwefel 2.22 and F2 Rosenbrock at the top; F3 Rastrigin and F4 Masters at the bottom

2. For Rosenbrock function (F2) whose global optimum is inside a long, narrow, parabolic-shaped flat valley. The convergence to the global optimum has been repeatedly used to assess the performance of optimization algorithm. MDE converges much faster than RAND and TDE in average since the beginning. In addition, MDE can successfully converge near to the optimum in some run when using NP=4D, thus resulting in the best value (at 1.67E-8) much better than all other algorithms.
3. For multimodal Rastrigin (F3) and the Masters (F4) functions whose fitness landscapes have numerous local optima due to the cosine term, MDE again outperforms other algorithms in all indicators but with only one exception. TDE can achieve the best values better than that by MDE in a few runs. However a lower SD value from MDE indicates MDE's better robustness than TDE.
4. Regarding the effects of different NP tested, in nearly all cases, all algorithms tested including MDE converge better, in average, when using NP=4D than NP=8D. The exception is only at multimodal Rastrigin case for MDE, in which a larger population tends to be more better than a smaller population. The effects of different population size and the guideline for choosing an optimal population size deserve a further investigation.

# 5 Financial Time-Series Prediction with SMN

This section describes an application of MDE in optimizing weights and biases of SMN during its learning, which is then used to predict the outputs during testing. The dataset is index of Dow Jones Industrial Average (DJIA) from 3 January 1994 to 23 October 2006, total samples is 3228 trading days. Raw data input are daily opening, high, low, closing, and volume traded, which are easily accessible online.

## 5.1 Training and Testing of the Prediction Model

The data are split into two sets – training and testing sets. The training set consists of 2510 patterns and the rest is set aside for testing [10]. In this experiment, six technical indicators reviewed of domain experts [9-10] are computed from the raw data as indicated in the Table 3 and then fed into the SMN. All the inputs are normalized to values between 0.1 and 0.9. Training of the prediction models is carried out using MDE algorithm to optimize six weights and six biases of the SMN, thus 12 decision (or objective) variables. The MDE optimization is to minimize the error, *E*, in eq. (2) where in this particular case, $d_p$ and $y_p$ are the actual and predicted indices, respectively, on day *p*. Due to stochastic nature of the algorithm, the optimized weight and bias values of the model are obtained through 5 independent runs with different seed number for random number generator. After each training run, the obtained set of six (weight, bias) pairs of the neuron is used for prediction of next-day closing index for the test data. The mean absolute percentage error (MAPE) in Eq (5) is used to gauge the performance of the prediction model when the testing data are used,

$$MAPE = \frac{1}{N_t}\sum_{p=1}^{N_t}\frac{|y_p - d_p|}{y_p}\times 100, \tag{5}$$

where $N_t$ is the number of testing data. Then basic statistical values of obtained MAPEs from 5 independent runs are computed for analysis. To compare the performance of the proposed model, other two DE variants (RAND and TDE) as well as back propagation (BP) learning algorithm (in Section 2.2) are also simulated. In this experiment, the learning period λ is 50 generations. Population size (NP) is 50. Maximum objective function calls (MAXNFC) allowed to learn is 30000 for MDE, TDE, and RAND, while standard BP for SMN is allowed to learn up to the same 30000 generations. For TDE, the mutation probability (*Mp*) is set to 0.05 as suggested by the originators [4]. Learning rate $\eta$ for BP is 0.7 [8-9]. Each of three DE variants is allowed to execute the same 5 independent runs and the statistical results (of MAPEs) are compared to the MAPE from the single run with SMN's BP.

**Table 3.** Selected technical indicators and their formula

| Technical indicators used | Formula |
|---|---|
| – Exponential moving average (EMA)<br>(3 numbers: EMA10, EMA 20, and EMA40) | (P×A)+(Prev. EMA×(1-A)); A = 2/(1+N), P is current price, A is smoothing factor, N is time period |
| – Accumulation/distribution oscillator (ADO) | ( (Close-Low) – (High-Close) ) / ( (High-Low) × (Period's Volume) ) |
| – Relative strength index (RSI) | RSI = 100 – 100/(1+U/D); U is total gain/n, D is total losses/n, n is number of RSI period |
| – Price rate of change (PROC) | $\frac{\text{(Today's close - Close X-period ago)}}{\text{Close X-period ago}} \times 100$ |

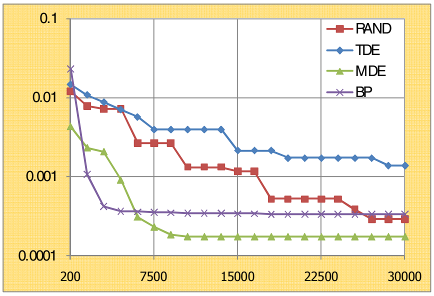


**Fig. 3.** Comparison of learning characteristic for DJIA from different algorithms

**Table 4.** Comparison of MAPE of DJIA during testing obtained from different algorithms

| Algorithm | Best | Average | Worst | S.D. |
|---|---|---|---|---|
| RAND | 1.72 | 22.04 | 74.51 | 30.02 |
| TDE | 19.92 | 31.36 | 34.59 | 6.40 |
| MDE | 1.61 | 1.69 | 1.86 | 0.11 |
| BP | 1.87 | | | |

## 5.2 Results and Discussions

The learning characteristics or convergences of four algorithms are demonstrated in Fig. 3. Table 4 compares the prediction accuracy using MAPE obtained from the algorithms. It can be observed that BP learning quickly converged and got trapped in some local optima early at MAPE=1.87. In contrary, RAND and TDE variants gradually improved their MAPEs similarly. However, RAND tends to provide a better result than TDE; this should due to the greediness of TDE that causes less exploration. A smaller value of *Mp* is expected to improve the results of TDE. Concerning the best MAPEs achieved, both RAND and MDE obtained the better values than BP.

However MDE achieved a better average MAPE than RAND. The small value of S.D. for MDE indicates its robustness in generalization for this prediction experiment.

## 6 Conclusion

Differential Evolution (DE) algorithm has recently gained a strong attention, and has been widely used in engineering and scientific optimization community. However its robustness and performance depend upon choosing a proper mutation strategy as well as associated control parameters. This work proposes the MDE algorithm that combines different efficient DE strategies to gain their advantages. The algorithm is evaluated using 5 well-known benchmark functions and compared to the standard DE/rand/1/bin and TDE variants. The results have demonstrated that MDE outperforms other competitive variants. Then MDE is applied to optimize both weights and biases of a single multiplicative neuron for prediction of the next-day closing index of DJIA. The results have shown that MDE can provide a better prediction in accuracy and robustness than back propagation, standard DE, and TDE. The current works are to experiment MDE with more strategies, and to evaluate with a larger set of benchmark functions and more time-series datasets.

## References

1. Storn, R., Price, K.: Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. J. of Global Optimization 11, 341–359 (1997)
2. Price, K., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series). Springer, New York (2005)
3. Chakraborty, U.K. (ed.): Advances in Differential Evolution. Springer, Heidelberg (2008)
4. Fan, H.Y., Lampinen, J.: A trigonometric mutation operation to differential evolution. Journal of Global Optimization 27, 105–129 (2003)
5. Yadav, R.N., Kalra, P.K., John, J.: Time series prediction with single multiplicative neuron model. Applied Soft Computing 7, 1157–1163 (2007)
6. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential Evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Evol. Comp. 13, 398–417 (2009)
7. Brest, J., Greiner, S., et al.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Trans. Evol. Comp. 10, 646–657 (2006)
8. Yadav, A., Mishra, D., Yadav, R.N., Ray, S., Kalra, P.K.: Time-series prediction with single integrate-and-fire neuron. Applied Soft Computing 7, 739–745 (2007)
9. Zhao, L., Yang, Y.: PSO-based single multiplicative neuron model for time series prediction. Expert Systems with Applications 36, 2805–2812 (2009)
10. Majhi, R., Panda, G., Majhi, B., Sahoo, G.: Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques. Expert Systems with Applications 36, 10097–10104 (2009)

# Boosted Neural Networks in Evolutionary Computation

Martin Holeňa[1], David Linke[2], and Norbert Steinfeldt[2]

[1] Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 18207 Prague
martin@cs.cas.cz

[2] Leibniz Institute for Catalysis, Albert-Einstein-Str. 29a, 18059 Rostock, Germany

**Abstract.** The paper deals with a neural-network-based version of surrogate modelling, a modern approach to the optimization of empirical objective functions. The approach leads to a substantial decrease of time and costs of evaluation of the objective function, a property that is particularly attractive in evolutionary optimization. In the paper, an extension of surrogate modelling with regression boosting is proposed, which increases the accuracy of surrogate models, thus also the agreement between results obtained with the model and those obtained with the original objective function. The extension is illustrated on a case study in materials science. Presented case study results clearly confirm the usefulness of boosting for neural-network-based surrogate models.

## 1 Introduction

For more than two decades, *evolutionary algorithms*, especially their most frequent representative – *genetic algorithms* (GAs), are successful in solving difficult optimization tasks [1,2]. Their popularity is to some extent due to the biological inspiration, which increases their comprehensibility outside computer science. Nevertheless, they have also important mathematical properties, such as the ability to escape a local optimum and continue the search for a global one, and the ability to do without information about gradients or second-order partial derivatives. This makes them particularly attractive for the optimization of *empirical objective functions*, the values of which cannot be mathematically calculated, but have to be obtained experimentally, through some measurement or testing. Indeed, the impossibility to calculate function values makes also calculation of derivatives impossible, whereas measurement errors usually hinder obtaining sufficiently accurate estimates of them.

Like other methods relying solely on function values, evolutionary algorithms need the objective function to be evaluated in a large number of points. In the context of optimization of empirical objective functions, this is quite disadvantageous because the evaluation of such a function is often costly and time-consuming. An area where the trade-off between successful optimization and costly objective function evaluations plays a crucial role is the computer-aided search for materials and chemicals optimal with respect to certain properties [3].

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 131–140, 2009.

Here, evolutionary algorithms are used in more than 90 % of optimization tasks, and the rarely encountered alternatives simulated annealing, simplex method and holographic search strategy also use solely function values, therefore needing a similarly high number of function evaluations. Testing a generation of materials or chemicals typically needs several days of time and costs thousands of euros. Therefore, evolutionary computation here rarely runs for more than 10 generations.

The usual approach to decreasing the cost and time of optimization of empirical objective functions is to evaluate the function only sometimes and to evaluate its suitable regression model otherwise. That model is termed *surrogate model* of the function, and the approach is referred to as *surrogate modelling*. Needless to say, the time and costs needed to evaluate a regression model are negligible. However, it must not be forgotten that the agreement between the results obtained with a surrogate model and those obtained with the original function depends on the accuracy of the model.

This paper suggests to increase the accuracy of surrogate models by means of boosting. Boosting is a popular approach to increasing the accuracy of classification, and due to its success in classification, also several methods of regression boosting have been proposed. However, so far no attempt has been reported to combine regression boosting with surrogate modelling. Hence, the purpose of the reported research is basically a proof of concept: to extend surrogate modelling with regression boosting, and to validate that extension on several sufficiently complex case studies. One of them is sketched in the paper.

In the following section, basic principles of surrogate modelling and its strategies in evolutionary optimization are recalled, and the importance of feed-forward neural networks as surrogate models is pointed out. Section 3 recalls a particular method of regression boosting and elaborates it for multilayer perceptrons. The application of such a boosted neural network as surrogate model is illustrated on a case study in materials science in Section 4.

## 2 Surrogate Models in Evolutionary Computation

Surrogate modelling is a general optimization approach in which the evaluation of the objective function is restricted to points considered to be most important for the progress of the employed optimization method [4,5,6,7]. It is most frequently encountered in connection with the optimization of empirical objective functions, but has been successfully applied also to expensive optimization tasks in engineering design in which the objective function is not empirical, but its evaluation is time-consuming [5].

Although surrogate modelling has been applied also to conventional optimization [4], it is most frequently encountered in connection with evolutionary algorithms because for them, the approach leads to the approximation of the fitness function, whose usefulness in evolutionary computation is already known [8,9]. For the progress of evolutionary optimization, most important criteria are on the one hand points that indicate closeness to the global optimum (through highest values of the fitness function), on the other hand points that most contribute to the diversity of the population.

In the literature, various possibilities of combining evolutionary optimization with surrogate modelling have been discussed [5,6,7]. Nevertheless, all of them are controlled by one of these two strategies:

A. The *individual-based-control* consists in choosing between the evaluation of the empirical objective function and the evaluation of its surrogate model individual-wise, for example, in the following steps:
   - **(i)** An initial set $\mathcal{E}$ of individuals in which the considered empirical fitness $\eta$ was evaluated (e.g., individuals forming several first generations of the evolutionary algorithm).
   - **(ii)** The surrogate model is constructed using pairs $\{(x, \eta(x)) : x \in \mathcal{E}\}$.
   - **(iii)** The evolutionary algorithm is run with the fitness $\eta$ replaced by the model for one generation with a population $\mathcal{Q}$ of size $qP$, where $P$ is the desired population size for the optimization of $\eta$, and $q$ is a prescribed ratio (e.g., $q = 10$ or $q = 100$).
   - **(iv)** A subset $\mathcal{P} \subset \mathcal{Q}$ of size $P$ is selected so as to contain those individuals from $\mathcal{Q}$ that are most important according to the considered criteria for the progress of optimization.
   - **(v)** For $x \in \mathcal{P}$, the empirical fitness is evaluated.
   - **(vi)** The set $\mathcal{E}$ is replaced by $\mathcal{E} \cup \mathcal{P}$ and control returns to the step (ii).

B. The *generation-based-control* consists in choosing between both kinds of evaluation generation-wise, for example, in the following steps:
   - **(i)** An initial set $\mathcal{E}$ of individuals in which the considered empirical fitness $\eta$ was evaluated is collected like with the individual-based control.
   - **(ii)** The surrogate model is constructed using pairs $\{(x, \eta(x)) : x \in \mathcal{E}\}$.
   - **(iii)** Relying on the error of the surrogate model, measured with a prescribed error measure (such as *mean squared error*, MSE, or *mean absolute error*, MAE), an appropriate number $g_m$ of generations is chosen, during which $\eta$ should be replaced by the model.
   - **(iv)** The evolutionary algorithm is run with the fitness $\eta$ replaced by the model for $g_m$ generations with populations $\mathcal{P}_1, \ldots, \mathcal{P}_{g_m}$ of size $P$.
   - **(v)** The evolutionary algorithm is run with the empirical fitness $\eta$ for a prescribed number $g_e$ of generations with populations $\mathcal{P}_{g_m+1}, \ldots, \mathcal{P}_{g_m+g_e}$ (frequently, $g_e = 1$).
   - **(vi)** The set $\mathcal{E}$ is replaced by $\mathcal{P} \cup \mathcal{P}_{g_m+1} \cup \cdots \cup \mathcal{P}_{g_m+g_e}$ and control returns to the step (ii).

For empirical objective functions, it is typical to be highly nonlinear. Hence, nonlinear regression models should be used as surrogate models. A prominent example are multilayer feed-forward neural networks, more precisely, the nonlinear mappings computed by such networks. Their attractiveness for nonlinear regression in general and for surrogate modelling in particular [8] is due to their *universal approximation capability*, which means that linear spaces of functions computed by certain families of multilayer feed-forward neural networks

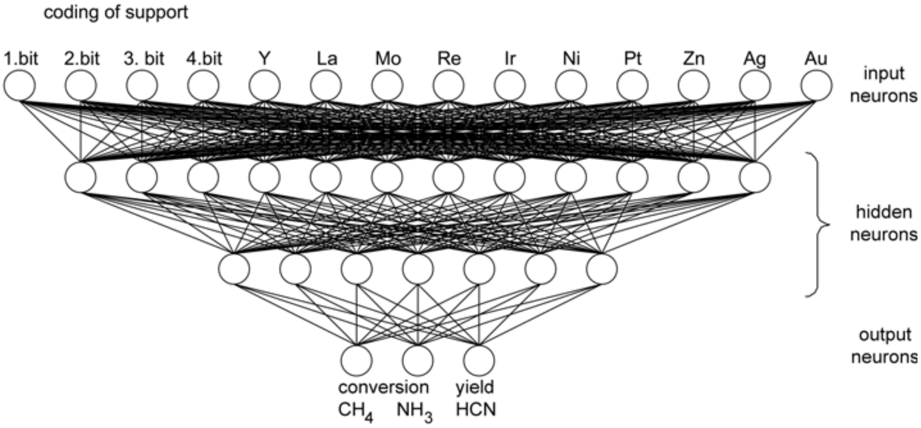


**Fig. 1.** Example multilayer perceptron architecture with two hidden layers, used in the case study presented in Section 4

are dense in some general function spaces [10,11]. In the application domain of catalytic materials, from which the case study in Section 4 is taken, nearly all examples of regression analysis published since mid 1990s rely on multilayer feed-forward neural networks, typically on multilayer perceptrons (MLPs, cf. Figure 1). In the last edition of "Handbook of heterogeneous catalysis", more than 20 such examples are listed, as well as several additional, based on radial basis function networks and piecewise-linear neural networks [12].

## 3 Boosted Neural Networks

Boosting was originally a method of improving classification accuracy through developing the classifier iteratively and increasing the relative influence of the training data that most contributed to errors in previous iterations on its development in subsequent iterations [13]. The usefulness of boosting for classification has incited its extension to regression [14,15]. In the following, the method *AdaBoost.R2*, proposed in [16], will be explained in detail, within an MLP context. Similarly to other adaptive boosting methods, each of the available input-output pairs $(x_1, y_1), \dots (x_p, y_p)$ is in the first iteration of AdaBoost.R2 used exactly once. This corresponds to resampling them according to the uniform probability distribution $P_1$ with $P_1(x_k, y_k) = \frac{1}{p}$ for $k = 1, \dots, p$. In addition, the weighted average error of the 1st iteration is set to zero, $\bar{E}_1 = 0$.

In the subsequent iterations ($i \geq 2$), the following steps are performed:

1. A sample $(\xi_1, \eta_1), \dots, (\xi_p, \eta_p)$ is obtained through resampling $(x_1, y_1), \dots$ $\dots, (x_p, y_p)$ according to the distribution $P_{i-1}$.
2. Using $(\xi_1, \eta_1), \dots, (\xi_p, \eta_p)$ as training data, a MLP with $n_I$ input neurons and $n_O$ output neurons is trained to compute a mapping $F_i : \mathbb{R}^{n_I} \to \mathbb{R}^{n_O}$.
3. A [0,1]-valued squared error vector $E_i$ of $F_i$ with respect to $(x_1, y_1), \dots$ $\dots, (x_p, y_p)$ is calculated as

$$E_i = (E_i(1), \ldots, E_i(p)) = \frac{(\|F_i(x_1) - y_1\|^2, \ldots, \|F_i(x_p) - y_p\|^2)}{\max_{k=1,\ldots,p} \|F_i(x_k) - y_k\|^2}, \qquad (1)$$

where $\| \quad \|$ denotes the Euclidean norm in $\mathbb{R}^{no}$.

4. The weighted average error of $F_i$ is calculated as $\bar{E}_i = \frac{1}{p}\sum_{k=1}^{p} P_i(x_k, y_k)E_i(k)$.
5. Provided $\bar{E}_i < 0.5$ , the probability distribution for resampling $(x_1, y_1), \ldots$ $\ldots, (x_p, y_p)$ is for $k = 1, \ldots, p$ updated according to

$$P_i(x_k, y_k) = \frac{P_{i-1}(x_k, y_k)\left(\frac{\bar{E}_i}{1-\bar{E}_i}\right)^{(1-E_i(k))}}{\sum_{k=1}^{p} P_{i-1}(x_k, y_k)\left(\frac{\bar{E}_i}{1-\bar{E}_i}\right)^{(1-E_i(k))}}. \qquad (2)$$

6. The *boosting approximation* in the $i$-th iteration is set to the componentwise median of the mappings $F_1, \ldots, F_i$ computed by the MLP up to the $i$-th iteration with respect to the probability distribution proportional to $\left(\frac{\bar{E}_1}{1-\bar{E}_1}, \ldots, \frac{\bar{E}_i}{1-\bar{E}_i}\right)$.

The errors used to asses the boosting approximation are called *boosting errors*, e.g., boosting MSE or boosting MAE. For simplicity, also the mapping $F_1$ computed by the MLP in the first iteration is called boosting approximation if boosting is performed, and the respective errors are called boosting errors, although boosting introduces no modifications in the 1st iteration.

The above formulation of the method deals only with the case $\bar{E}_i < 0.5$. For $\bar{E}_i \geq 0.5$, the original proposal of the method in [16] stops boosting. However, that is not allowed if the stopping criterion should be based on an independent set of validation data. Indeed, the calculation of $\bar{E}_i$ does not rely on independent data, but it relies solely on the data employed to construct the $F_i$. A possible alternative for the case $\bar{E}_i \geq 0.5$ is reinitialization, proceeding as in the 1st iteration [17].

In connection with surrogate modelling, two remarks to boosted neural networks are appropriate:

A. Boosted neural networks (more generally, any boosted surrogate models) are only particular kinds of surrogate models and their interaction with evolutionary algorithms in optimization tasks follows the same rules as the interaction of surrogate models in general. In particular in the outlines of individual-based and generation-based control in Section 2, boosting is always performed in the step (i).

B. It is important to be aware of the difference between the iterations of boosting and the iterations of neural network training. Boosting iterates on a higher level, one iteration of boosting includes a complete network training, which can proceed for hundreds of iterations. Nevertheless, both kinds of iterations are similar in the sense that starting with a certain iteration, over-training is present. Therefore, also over-training due to boosting can be reduced through *stopping* in the iteration after which the *error on an independent set of validation data increases*. Moreover, *cross-validation* can be used to find the iteration most appropriate for stopping.

## 4 Case Study in Materials Science

The extension of MLP-based surrogate modelling with boosting will be illustrated on a case study with data from the investigation of *catalytic materials for the high-temperature synthesis of hydrocyanic acid* [18]. It was performed through experiments in a circular 48-channel reactor. In most of those experiments, the composition of the materials was designed by means of a *specific GA for heterogeneous catalysis* [19]. As usually in evolutionary optimization of catalytic materials, the GA configuration was determined by the experimental conditions in which the optimization was performed: number of channels of the reactor in which the materials were tested, as well as time and financial resources available for those expensive tests. In the reported case study, the algorithm was running for 7 generations of population size 92, and in addition 52 other catalysts with manually designed composition were investigated. Consequently, data about 696 catalytic materials were collected.

The MLPs employed as surrogate models had *14 input neurons*: 4 of them coding the material used as support, the other 10 corresponding to the proportions of 10 metal additives belonging to independent variables; output neurons were 3, corresponding to considered fitness functions (Figure 1).

The most appropriate MLP architectures were searched with cross-validation, using only data about catalysts from the 1.–6. generation of the GA and about the 52 catalysts with manually designed composition, thus altogether data about 604 catalytic materials. To use as much information from data as possible, cross-validation was applied as the extreme 604-fold variant, leave-1-out validation. The set of architectures within which the search was performed was delimited by means of the heuristic *pyramidal condition*: the number of neurons in a subsequent layer must not increase the number of neurons in a previous layer.

To investigate the usefulness of boosting in our case study, the same data were used and the same set of architectures was considered as for architecture search. In each iteration of boosting, a leave-1-out validation was performed, and the boosting MSE calculated in each fold for the catalytic material left out from training the respective MLP was subsequently averaged over all 604 folds. The *criterion* according to which *boosting is considered useful* for an architecture was: the average boosting MSE in the 2nd iteration has to be lower than in the 1st iteration. The iteration till which the average boosting MSE continuously decreased was then taken as the *final iteration of boosting*.

According to that criterion, boosting was useful for 9 among the 12 considered architectures with one hidden layer and for 65 among the 78 considered architectures with two hidden layers. For those architectures, basic information relevant to boosting is summarized in Table 1. In particular, the difference of the average

**Table 1.** Architectures for which boosting improved the boosting MSE on the test data, averaged over the leave-1-out validation, together with the final iteration till which it was improved, and together with values of boosting MSE $[\cdot 10^{-3}]$ in the 1st and final iteration

| architecture | final iteration | boosting MSE, 1st iteration | boosting MSE, final iteration | architecture | final iteration | boosting MSE, 1st iteration | boosting MSE, final iteration | architecture | final iteration | boosting MSE, 1st iteration | boosting MSE, final iteration |
|---|---|---|---|---|---|---|---|---|---|---|---|
| (14,3,3) | 2 | 8.89 | 8.88 | (14,8,5,3) | 5 | 8.54 | 7.91 | (14,12,8,3) | 6 | 8.13 | 7.85 |
| (14,4,3) | 7 | 8.99 | 8.86 | (14,8,6,3) | 14 | 9.23 | 8.34 | (14,12,9,3) | 7 | 9.91 | 9.10 |
| (14,5,3) | 3 | 8.17 | 7.11 | (14,8,8,3) | 2 | 7.66 | 7.53 | (14,12,10,3) | 2 | 8.47 | 8.40 |
| (14,6,3) | 3 | 7.80 | 7.75 | (14,9,3,3) | 8 | 10.24 | 7.60 | (14,12,12,3) | 13 | 8.62 | 7.31 |
| (14,8,3) | 2 | 7.22 | 7.20 | (14,9,4,3) | 2 | 9.06 | 8.58 | (14,13,3,3) | 9 | 11.43 | 9.30 |
| (14,9,3) | 2 | 7.95 | 7.92 | (14,9,5,3) | 10 | 8.96 | 8.13 | (14,13,4,3) | 2 | 10.08 | 8.70 |
| (14,11,3) | 3 | 7.13 | 7.11 | (14,9,6,3) | 8 | 10.94 | 10.08 | (14,13,5,3) | 31 | 9.55 | 6.74 |
| (14,13,3) | 2 | 12.11 | 12.03 | (14,9,7,3) | 5 | 9.09 | 8.74 | (14,13,6,3) | 20 | 9.55 | 7.27 |
| (14,14,3) | 3 | 7.38 | 7.35 | (14,10,3,3) | 8 | 10.08 | 9.00 | (14,13,7,3) | 7 | 9.29 | 8.59 |
| | | | | (14,10,4,3) | 19 | 9.63 | 7.07 | (14,13,8,3) | 3 | 8.68 | 8.14 |
| (14,3,3,3) | 2 | 8.78 | 8.67 | (14,10,5,3) | 3 | 8.78 | 8.35 | (14,13,9,3) | 8 | 9.52 | 8.19 |
| (14,4,3,3) | 19 | 8.74 | 7.50 | (14,10,6,3) | 32 | 8.69 | 6.55 | (14,13,10,3) | 3 | 8.44 | 8.22 |
| (14,4,4,3) | 11 | 9.25 | 8.67 | (14,10,7,3) | 4 | 8.63 | 8.04 | (14,13,12,3) | 13 | 9.08 | 7.60 |
| (14,5,3,3) | 5 | 8.94 | 8.16 | (14,10,10,3) | 3 | 9.70 | 9.41 | (14,13,13,3) | 13 | 8.39 | 7.41 |
| (14,5,5,3) | 3 | 8.28 | 7.92 | (14,11,3,3) | 7 | 11.30 | 9.03 | (14,14,3,3) | 6 | 11.52 | 8.89 |
| (14,6,3,3) | 8 | 9.16 | 8.46 | (14,11,5,3) | 8 | 9.81 | 7.98 | (14,14,4,3) | 15 | 9.56 | 7.13 |
| (14,6,4,3) | 3 | 8.55 | 8.22 | (14,11,6,3) | 6 | 8.69 | 7.82 | (14,14,5,3) | 7 | 10.43 | 8.82 |
| (14,6,6,3) | 3 | 7.22 | 7.16 | (14,11,7,3) | 7 | 9.59 | 8.92 | (14,14,7,3) | 7 | 9.04 | 7.87 |
| (14,7,3,3) | 5 | 8.54 | 7.70 | (14,11,8,3) | 24 | 9.18 | 7.53 | (14,14,8,3) | 29 | 9.73 | 6.65 |
| (14,7,4,3) | 5 | 9.24 | 8.22 | (14,11,10,3) | 2 | 9.28 | 9.13 | (14,14,9,3) | 3 | 9.14 | 8.48 |
| (14,7,5,3) | 2 | 8.58 | 8.29 | (14,12,3,3) | 11 | 10.94 | 7.31 | (14,14,10,3) | 4 | 9.02 | 8.18 |
| (14,7,6,3) | 5 | 7.75 | 7.56 | (14,12,4,3) | 13 | 10.80 | 8.46 | (14,14,11,3) | 10 | 8.36 | 7.96 |
| (14,7,7,3) | 4 | 8.27 | 7.27 | (14,12,5,3) | 8 | 8.97 | 8.32 | (14,14,12,3) | 3 | 8.84 | 8.54 |
| (14,8,3,3) | 17 | 10.47 | 8.18 | (14,12,6,3) | 6 | 9.18 | 8.71 | (14,14,13,3) | 3 | 9.36 | 9.06 |
| (14,8,4,3) | 6 | 11.06 | 9.96 | (14,12,7,3) | 7 | 10.11 | 9.17 | (14,14,14,3) | 6 | 8.77 | 8.18 |

boosting MSE between the first and final iteration is a measure of accuracy improvement of the boosted neural network compared to the classical neural network trained in the 1st iteration. Moreover, boosting has not improved the accuracy of neural networks with architectures not listed in Table 1 because for them, the average boosting MSE in the 2nd iteration was higher than the average MSE of the classical neural network trained in the 1st iteration.

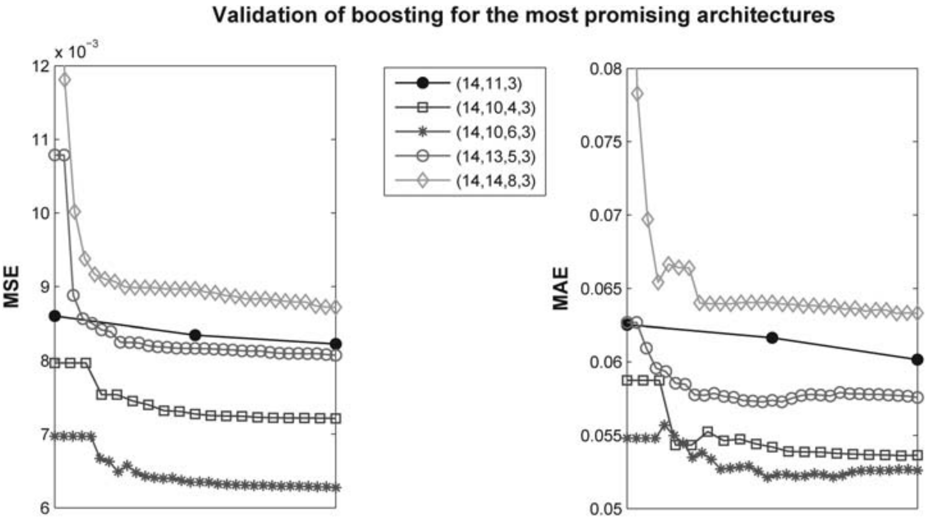


**Fig. 2.** History of the boosting MSE and MAE on the data from the 7th generation of the GA for MLPs with the 5 architectures included in the validation of boosting

To validate the most promising results of the investigation of the usefulness of boosting in our case study, data from the 7th generation of the genetic algorithm were used. The validation included the *5 architectures* that were most promising for boosting from the point of view of the *lowest boosting MSE on the cross-validation test data in the final iteration*. According to Table 1, these were the architectures (14,11,3), (14,10,4,3), (14,10,6,3) (14,13,5,3) and (14,14,8,3) for which the final iterations of boosting were 3, 19, 32, 31 and 29 respectively. For each of them, the validation proceeded as follows:

1. In each iteration up to the final, a single MLP was trained with data about the 604 catalytic materials considered during the architecture search.
2. In each iteration, the boosting approximations of the MLP outputs were computed.
3. From the values predicted by the boosting approximations for the 92 materials from the 7. generation of the GA, and from the measured values, the boosting MSE and MAE were calculated.

The results are summarized in Figure 2. They clearly confirm the usefulness of boosting for the 5 considered architectures. For each of them, boosting led to an overall decrease of both considered error measures, MSE and MAE, on new data from the 7th generation of the GA. Moreover, the decrease of the MSE (which is the measure employed during the investigation of the usefulness of boosting) was uninterrupted or nearly uninterrupted till the final iteration of boosting.

## 5 Conclusions

The paper dealt with surrogate modelling, a modern optimization approach, which is particularly attractive in evolutionary optimization of empirical objective functions. It proposed to extend surrogate modelling with regression boosting, to increase the accuracy of surrogate models, thus also the agreement between results obtained with a surrogate model and those obtained with the original objective function. The proposed extension was elaborated for the important situation when a function computed by a MLP is employed as surrogate model. Needless to say, regression boosting is not new, though it is less common than the popular classification boosting. However, novel is its combination with surrogate models, which adds the advantage of increased accuracy to the main advantage of surrogate modelling – decreasing the time and costs of optimization.

Theoretical principles of both surrogate modelling and boosting are known, therefore the main purpose of the reported research was to validate the feasibility of the proposed extension of surrogate modelling on several sufficiently complex case studies, one of which was sketched in this paper. The presented case study results clearly confirm the usefulness of boosting. From the point of view of the average boosting MSE, the performance of boosted neural networks was better than the performance of classical neural networks for 74 tested architectures, whereas the opposite situation occurred only for 16 tested architectures. For the five most promising architectures, boosting led to an overall decrease of MSE and MAE on new data from the last generation of the evolutionary algorithm. If boosting error is measured in the same way as during the investigation of the usefulness of boosting, i.e. as MSE, then the decrease was uninterrupted or nearly uninterrupted till the final iteration.

Notice that the error measures MSE and MAE, which were employed in the case study, take into account only the accuracy of the approximations of the empirical objective functions by the surrogate models, and not the complexity of the models. Therefore, one of the main objectives of our future research into boosted surrogate models is testing the applicability of the Bayesian information criterion and the Akaike information criterion in this context. These error measures take into account model complexity, thus allowing to tune the trade-off between the accuracy of the neural network approximation and its smoothness. That testing will be a part of a comprehensive series of tests of boosted neural networks and their comparison with classical neural networks on several artificial and real-world data sets.

## Acknowledgment

The reported research has been supported by the grants 201/08/0802 as well as ICC/08/E018 of the Grant Agency of the Czech Republic.

## References

1. Reeves, C., Rowe, J.: Genetic Algorithms: Principles and Perspectives. Kluwer Academic Publishers, Boston (2003)

2. Schaefer, R.: Foundation of Global Genetic Optimization. Springer, Heidelberg (2007)
3. Baerns, M., Holeňa, M.: Combinatorial Development of Solid Catalytic Materials. Design of High-Throughput Experiments, Data Analysis, Data Mining. World Scientific, Singapore (2009)
4. Brooker, A., Dennis, J., Frank, P., Serafini, D.V.T., Trosset, M.: A rigorous framework for optimization by surrogates. Structural and Multidisciplinary Optimization 17, 1–13 (1998)
5. Ong, Y., Nair, P., Keane, A., Wong, K.: Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems. In: Jin, Y. (ed.) Knowledge Incorporation in Evolutionary Computation, pp. 307–331. Springer, Heidelberg (2005)
6. Ulmer, H., Streichert, F., Zell, A.: Model assisted evolution strategies. In: Jin, Y. (ed.) Knowledge Incorporation in Evolutionary Computation, pp. 333–355. Springer, Heidelberg (2005)
7. Zhou, Z., Ong, Y., Nair, P., Keane, A., Lum, K.: Combining global and local surrogate models to accellerate evolutionary optimization. IEEE Transactions on Systems, Man and Cybernetics. Part C: Applications and Reviews 37, 66–76 (2007)
8. Jin, Y., Hüsken, M., Olhofer, M., Sendhoff, B.: Neural networks for fitness approximation in evolutionary optimization. In: Jin, Y. (ed.) Knowledge Incorporation in Evolutionary Computation, pp. 281–306. Springer, Heidelberg (2005)
9. Ratle, A.: Accelerating the convergence of evolutionary algorithms by fitness landscape approximation. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 87–96. Springer, Heidelberg (1998)
10. Hornik, K.: Approximation capabilities of multilayer neural networks. Neural Networks 4, 251–257 (1991)
11. Pinkus, A.: Approximation theory of the MPL model in neural networks. Acta Numerica 8, 277–283 (1998)
12. Ertl, G., Knözinger, H., Schüth, F., Eitkamp, J. (eds.): Handbook of Heterogeneous Catalysis. Wiley-VCH, Weinheim (2008)
13. Schapire, R.: The strength of weak learnability. Machine Learning 5, 197–227 (1990)
14. Friedman, J.: Greedy function approximation: A gradient boosting machine. Annals of Statistics 29, 1189–1232 (2001)
15. Shrestha, D.: Experiments with AdaBoost.RT, an improved boosting scheme for regression. Neural Computation 18, 1678–1710 (2006)
16. Drucker, H.: Improving regression using boosting techniques. In: Sharkey, A. (ed.) Proceedings of the 14th International Conference on Machine Learning, pp. 107–115. Springer, Heidelberg (1997)
17. Altinçay, H.: Optimal resampling and classifier prototype selection in classifier ensembles using genetic algorithms. Pattern Analysis and Applications 7, 285–295 (2004)
18. Möhmel, S., Steinfeldt, N., Endgelschalt, S., Holeňa, M., Kolf, S., Dingerdissen, U., Wolf, D., Weber, R., Bewersdorf, M.: New catalytic materials for the high-temperature synthesis of hydrocyanic acid from methane and ammonia by high-throughput approach. Applied Catalysis A: General 334, 73–83 (2008)
19. Wolf, D., Buyevskaya, O., Baerns, M.: An evolutionary approach in the combinatorial selection and optimization of catalytic materials. Applied Catalyst A: General 200, 63–77 (2000)

# Improving Prediction Interval Quality: A Genetic Algorithm-Based Method Applied to Neural Networks

Abbas Khosravi, Saeid Nahavandi, and Doug Creighton

Centre for Intelligent Systems Research (CISR)
Deakin University, Geelong, Australia
{akhos,saeid.nahavandi,doug.creighton}@deakin.edu.au

**Abstract.** The delta technique has been proposed in literature for constructing prediction intervals for targets estimated by neural networks. Quality of constructed prediction intervals using this technique highly depends on neural network characteristics. Unfortunately, literature is void of information about how these dependences can be managed in order to optimize prediction intervals. This study attempts to optimize length and coverage probability of prediction intervals through modifying structure and parameters of the underlying neural networks. In an evolutionary optimization, genetic algorithm is applied for finding the optimal values of network size and training hyper-parameters. The applicability and efficiency of the proposed optimization technique is examined and demonstrated using a real case study. It is shown that application of the proposed optimization technique significantly improves quality of constructed prediction intervals in term of length and coverage probability.

**Keywords:** Neural network, genetic algorithm, prediction interval.

## 1 Introduction

A generic neural network model is a non-parametric attempt to model the human brain. Composed of many parallel, interconnected computing units, NNs can identify and learn the nonlinear dependencies amongst the sets of inputs and outputs. With their flexible mathematical structures, they have ability to approximate any nonlinear mapping with any arbitrary degree of accuracy (universal approximator). Relying on these promising features, researchers have widely applied them in many real-world problems, including engineering regression and classification problems. Their superiority over traditional parametric techniques/models has been reported frequently [1] [2].

In the majority of studies and research conducted so far, focus is on developing NN models for point prediction, without any indication of its likely accuracy. As systems become more and more complex with many interconnected components, uncertainty creeps into their operation. Accompanied with uncertainty are probabilistic events whose occurrence may result in multiple realities for future of a system. These issues negatively affect performance of the NN regression models for predicting future of the underlying systems. Degradation of NN prediction performance is more tangible when targets are from complex systems including, among others, manufacturing enterprises, transportation systems, or airports. One should make note of the fact that

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 141–149, 2009.

**Nomenclature**

All vectors are column vectors and are denoted by boldface letters.

| | |
|---|---|
| $x_i$ | The i-th nonrandom m-input vector |
| $\mathcal{D}_1$ & $\mathcal{D}_2$ | First and second training datasets |
| $\mathcal{D}_{test}$ | Test dataset |
| $\mathbf{W}$ | Original set of neural network parameters |
| $\boldsymbol{w}_{opt}$ | Optimized set of neural network parameters |
| $\mathbf{w}^*$ | the true set of NN parameters |
| $\hat{\mathbf{w}}$ | the set of NN parameters obtained using least square technique |
| $\hat{y}_i$ | Neural network scalar output depending on |
| $\Phi(x_i, \mathbf{w}^*)$ | Nonlinear function of parameterized by |
| $\delta^2$ | Variance of |
| $\nabla_w^T y$ | Gradient of neural network output with respect to its set of parameters |
| $J$ | Jacobian matrix of neural network |
| $s$ | The standard deviation estimate |
| $\varepsilon_i$ | The i-th error term associated with modelling function |
| $\lambda$ | Regularizer constant |
| $t_d^{1-\frac{\alpha}{2}}$ | $1-\frac{\alpha}{2}$ quantile of a cumulative t-distribution function with d degrees of freedom |
| $\xi$ | Range of the underlying target |
| $\mathbb{N}^*$ | Set of positive natural numbers |
| $\mathbb{R}$ | Set of positive natural numbers |
| $n_i$ | Number of neurons in each layer of a NN with $L$ layer |
| $n_i^{upper}$ | Upper bound of $n_i$ |
| MSE | Mean Squared Error |

negative consequences raised from the stochastic nature of complex systems cannot be compensated solely through increasing NN size (neither hidden layers nor neurons) or repeating its training procedure. Because NN regression models offer a point estimate without any measure of its accuracy, making decisions based on such prediction may spell disaster for different components of large systems.

To cope with these deficiencies, construction of Prediction Intervals (PIs) for NN outputs has been proposed in literature. Mathematically, a prediction interval is a random interval that future observations will lie within it with a previously determined portability called confidence level. In literature, a variety of methods (from different perspectives) has been proposed and examined for constructing PIs for NNs [3] [4] [5] [6] [7] [8]. The focus of this research is on delta technique for constructing prediction intervals [3] [4]. Delta technique has its roots in nonlinear regression [9]. It interprets and represents NN models as nonlinear regression models. Such interpretation makes it possible to apply standard asymptotic theory to them for constructing PIs. Based on this method, a NN model can be interpreted as a nonlinear mapping,

$$y_i = \Phi(\mathbf{x}_i, \mathbf{w}^*) + \varepsilon_i i = 1,2, \dots, m \tag{1}$$

where $\varepsilon_i$ are assumed to be independently and identically distributed (iid) with variance$\delta^2$.The Taylor series expansion of (1) produces the following estimate, $\hat{y}_i$,

$$\hat{y}_i = \Phi(\mathbf{x}_i, \hat{\mathbf{w}}) = \Phi(\mathbf{x}_i, \mathbf{w}^*) + \nabla_{\mathbf{w}^*}^T y \, (\hat{\mathbf{w}} - \mathbf{w}^*) \tag{2}$$

According to theories of nonlinear regression, a $(1-\alpha)\%$ asymptotic prediction interval for $\hat{y}_i$ will be as follows,

$$\hat{y}_i \pm t_d^{1-\frac{\alpha}{2}} s \left[1 + \nabla_{w^*}^T y \, (J^T J)^{-1} \nabla_{w^*} y\right]^{\frac{1}{2}} \quad i = 1,2, \dots, m \tag{3}$$

When networks are over-fitted, the Jacobian matrix becomes singular. In case of using a weight decay regularizer to avoid such a problem ($(y - \hat{y})^T (y - \hat{y}) + \lambda \, w^T w$), the PIs will be constructed as follows [4],

$$\hat{y}_i \pm t_d^{1-\frac{\alpha}{2}} s \left[1 + \nabla_{w^*}^T y_i (J^T J + \lambda I)^{-1} (J^T J)(J^T J + \lambda I)^{-1} \nabla_{w^*} y_i\right]^{\frac{1}{2}} \quad i = 1,2, \dots, m. \tag{4}$$

Researchers have used (3) and (4) for constructing PIs in different case studies [6,10].

Despite reports on successful applications of the delta technique, there are many issues left unarticulated in this domain. One issue, which is in fact the main motivation for conducting this research, is that how PIs can be constructed to have the minimum length with the highest coverage probability. Coverage probability of PIs has a direct relationship with length of PIs: the longer the PIs are, the higher the coverage probability is. The perfect coverage probability can be easily achieved through considering extreme values of targets as upper and lower bounds of PIs. Needless to say, such wide PIs are useless, as they carry no information about targets' variation. Based on this argument, this paper aims at optimizing length and coverage probability of PIs: squeezing intervals without compromising their coverage probability. The optimization algorithm attempts to select the optimal structure for NN and determine optimal values for some parameters leading to narrower PIs without reduction in their coverage.

The rest of this paper is organized as follows. Section 2 of this paper explains how PI construction can be cast in the format of an optimization problem. Section 3 describes the optimization method and its practical issues. Results for a case study are demonstrated and discussed in Section 4. Section 5 concludes the paper with a short summary and some guidelines for further research.

## 2 Problem Formulation

Quality of PIs constructed using (4) depends on many factors. Of the significantly influential factors are NN size and parameters. Network size indicates number of adaptive parameters of NNs which is a function of number of inputs, number of layers, and number of neurons per layer (in case of a multi-input single-output (MISO) network). The direct effects of network size is on Jacobian matrix which appears in (4) in the form of $(J^T J + \lambda I)^{-1} (J^T J)(J^T J + \lambda I)^{-1}$. Dimensionality of the Jacobian matrix is determined by number of network parameters. The Jacobian matrix of NN with bigger size has more elements that do smaller networks. Therefore, under equal conditions, wideness of PIs is partially attributable to the dimensionality of the Jacobian matrix.

Length of PIs in (4) also depends on estimation of standard deviation, s. In order to minimize PI lengths, one solution is to minimize s as much as possible. While this minimization may positively contribute to squeezing PIs, it may lead to a serious problem: over-fitting. In such a case, the gradient terms in (4) evaluated for not-yet-seen samples will quickly jump up. This again will significantly widen PIs. Over-fitting problem can be handled through increasing the value of the regulizer factor (λ). Unfortunately, this again will result in wider intervals due to its direct presence in (4).

A satisfactory tradeoff between the aforementioned conflicting issues can be achieved through implementation of an optimization problem. In mathematical terms, the optimization problem can be written as follows (for an L layer NN),

$$\min_{n_1, n_2, \lambda} \text{NMPIL} \tag{5}$$

Subject to,

$$PICP \geq 1 - \alpha - \frac{\alpha}{\gamma} \tag{6}$$

$$n_i \text{and } n_i^{\text{upper}} \in \mathbb{N}^* \qquad i = 1,2,\ldots, L \tag{7}$$

$$1 \leq n_i \leq n_i^{\text{upper}} \qquad i = 1,2,\ldots, L \tag{8}$$

$$0 \leq \lambda \leq 1, \qquad \lambda \in \mathbb{R} \tag{9}$$

Normalized Mean Prediction Interval Length (NMPIL) is a measure of length of PIs normalized by the underlying target range,

$$\text{NMPIL} = \frac{1}{m} \sum_{i=1}^{m} \frac{2\, t_d^{1-\frac{\alpha}{2}}\, s\, [1 + \nabla_w^T y_i (J^T J + \lambda I)^{-1} (J^T J)(J^T J + \lambda I)^{-1} \nabla_w\, y_i]^{\frac{1}{2}}}{\xi} \tag{10}$$

Prediction Interval Coverage Probability (PICP) is also computed as follows,

$$\text{PICP} = \frac{1}{m} \sum_{i=1}^{m} c_i \tag{11}$$

where $c_i = 1$, if the target is covered by PI, otherwise $c_i = 0$. Constraint (6) guarantees that PICP always remain sufficiently high during optimization process. $\gamma$ is a constant (bigger than one) determined by modeler. Constraint (7) is related to the number of neurons in different layers of NN model. In (7), an upper bound has been considered for number of neurons in each layer. Such an upper bound can be determined empirically based on the principle that big NNs are highly over-fitted. Term (9) constrains range and domain of the regulizer factor.

## 3 Optimization Algorithm

In this section, the optimization method for minimizing (5) while satisfying constraints (6)-(9) is described. As some optimization variables are integers, the optimization function is not continuous and differentiable. Therefore, minimization of (5) through mathematical analysis is not possible. Even if possible, because mathematical

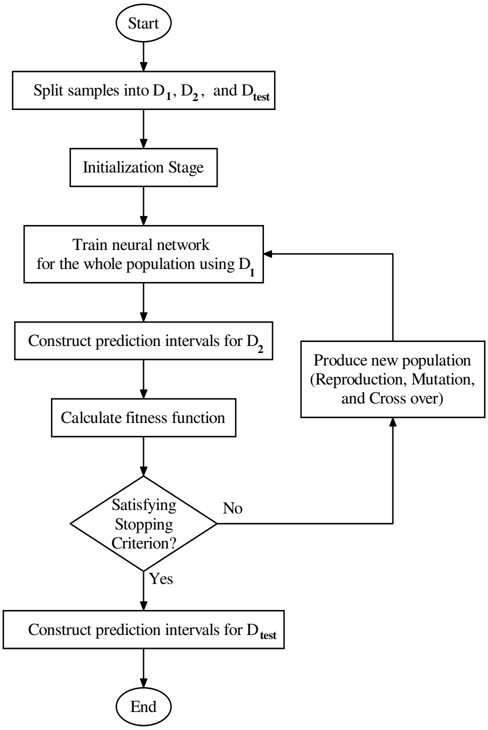

Start
Split samples into $D_1$, $D_2$, and $D_{test}$
Initialization Stage
Train neural network for the whole population using $D_1$
Construct prediction intervals for $D_2$
Calculate fitness function
Satisfying Stopping Criterion?
No
Yes
Produce new population (Reproduction, Mutation, and Cross over)
Construct prediction intervals for $D_{test}$
End

**Fig. 1.** The proposed method for developing optimal prediction intervals

optimization does not explore the parameter space, it is highly likely that they end up trapped in local minima. Unfortunately, local solutions are unsatisfactory for problems where the variation of the fitness function is not monotonous with regard to the parameters, as it is in our case. Due to these characteristics, evolutionary techniques such as Genetic Algorithm (GA) are the most appropriate search and optimization tools. Essentially, GA is a stochastic search algorithm inspired by the mechanics of natural evolution including survival of the fittest, reproduction, cross-over and mutation. The most interesting feature of GA is that it does not evaluate and improve a single solution, but analyses and modifies a set of solutions concurrently. The efficacy of GA for solving optimization problems and its superiority has been demonstrated in literature [11,12]. In literature, GA has been widely used for finding optimal structures of NNs with the purpose of minimizing the prediction error [13] [14].

With regard to the powerful features of GA and conditions of the underlying problem, GA is adopted for finding the optimal structure and regulizer factor. The optimization mechanism of the proposed method can be succinctly described as follows. Firstly, data samples are split into three datasets: training dataset 1 ($\mathcal{D}_1$), training data set 2 ($\mathcal{D}_2$), and the test set ($\mathcal{D}_{\text{test}}$). $\mathcal{D}_1$ and $\mathcal{D}_2$ are used for training NNs and constructing PIs. $\mathcal{D}_{\text{test}}$ is for evaluation of the optimized NN using not-yet-seen data. An initial population is considered for parameters of the interest (number of neurons in first and second layer, and the regulizer factor). In the GA loop, Levenberg–Marquardt method is applied for training NN models using $\mathcal{D}_1$. Training is repeated for the whole population of candidates. Then, PIs are constructed for $\mathcal{D}_2$ based on the delta technique. The fitness function (NMPIL) is computed for each case. Term s in (10) is calculated for $\mathcal{D}_1$. If the stopping criteria of the GA are unsatisfied, a new population will be generated and the GA loop will be repeated. Exploration of candidate space continues until some termination criteria are met. After termination of the GA loop, PIs for $\mathcal{D}_{\text{test}}$are constructed using the optimized NN model.

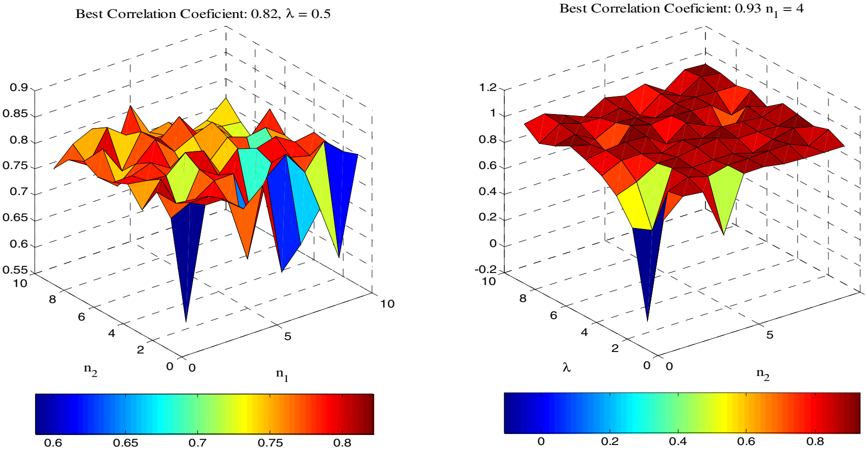


**Fig. 2.** Correlation coefficients between point prediction and targets, (left-side), varying number of neurons while keeping the regulizer factor fixed, (right-side), considering 4 neurons in first layer and varying other parameters

Fig. 1 shows the flowchart of the proposed GA based optimization method for finding the optimal values of $n_1, n_2$, and $\lambda$. The GA termination criterion can be reaching a maximum number of iterations, achieving a minimum fitness limit, or exceeding a stall limit.

## 4 Experiments and Numerical Results

In this section, numerical tests are conducted to evaluate usefulness and effectiveness of the proposed optimization method for minimization of PI length through finding the best structure for the NN models and the optimal value for the regulizer factor used in (4). The case study is a benchmark dataset frequently used in literature [15]. It contains thirteen independent variables and one dependent variable (house price). Total number of samples is 506. In the experiment, $\mathcal{D}_1$ and $\mathcal{D}_2$ and $\mathcal{D}_{test}$contain 50%, 25%, and 25% of total samples, respectively. All PIs are constructed with 90% confidence.$\gamma$ in (6) has been considered to be 2.The optimization is carried out with a population size of 30 and the stopping criterion as the maximum number of generations, which is set to 50. Optimization terminates if there is no improvement for 15

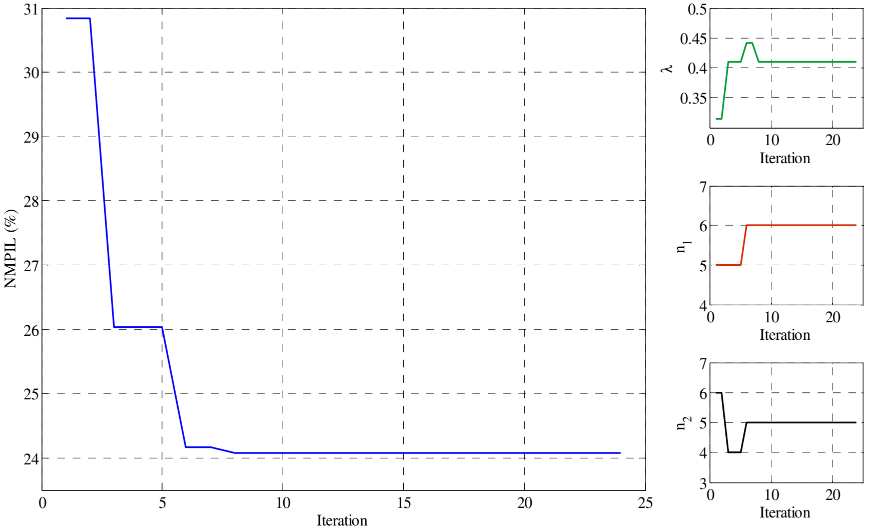


**Fig. 3.** NMPIL evolution through optimization procedure (left-side), and optimal values of parameters (right-side)

generations. To avoid any loss of good solutions, the allowed range for $n_1$ and $n_2$ is considered to be $[1,15]$. After termination of optimization, the final network is used for constructing optimal PIs for the unobserved testing samples ($\mathcal{D}_{test}$).

It is first shown that point prediction error is always big, regardless of network structure and the regulizer factor value. Fig. 2 represents correlation coefficient calculated for NN estimation and real targets. In the experiment correspond to the left-side plot, number of neurons in the first and second layer has been varied between 1 to 10, while keeping the regulizer factor fixed at 0.5.In another experiment (the right-side plot), quantity of neurons in the second layer and the regulizer factor have been varied, while keeping quantity of neurons in the second layer fixed at 4. To avoid problems related to initialization, training procedure has been repeated 5 times and average results have been reported. The correlation coefficient maximums for two experiments are 0.82 and 0.93, respectively. Because these coefficients have a considerable difference with the perfect case, NN point predictions are not much reliable. Results clearly indicate that this unreliability is not compensable through changing NN structure or parameters.

PIs are constructed using the delta technique and optimized based on the proposed method in the previous sections. Variation of parameters and the fitness function during optimization have been represented in Fig. 3. Optimization terminates after 24 generations. The optimal values for $\mathrm{n}_1, \mathrm{n}_2$, and λ are 5, 6, and 0.41, respectively. The fitness function quickly drops in the early generations of GA and then plagues until optimization terminates. The obtained results clearly indicate that optimization of NN structure and the regulizer factor has a strong impact on length of PIs.

The real value of any optimization problem must be verified using unobserved samples. Here, samples of $\mathcal{D}_{\mathrm{test}}$ are projected to the optimized NN. Table 1 represents

NMPIL and PICP for the first and final (optimized) NNs. These results show that the optimization procedure yields considerably narrower PIs (23.05%) without compromising coverage probability of intervals (88.2%). Besides, provided results explicitly indicate that generalization capability of NN models has been maintained and even improved through the optimization procedure. This is mainly due to the fact that two sets have been used during the optimization process.

**Table 1.** NMPIL and PICP for test samples ($\mathcal{D}_{test}$)

| NN Model | NMPIL (%) | PICP (%) |
|---|---|---|
| Obtained in first generation of GA | 29.49 | 93.70 |
| Optimized NN | 23.05 | 88.20 |

## 5 Conclusion

The first conclusion of this paper is that quality of prediction interval is highly affected by neural network structure and training hyper-parameters. Length and coverage probability of prediction intervals highly depend on neural network structure and the regulizer factor. Therefore, optimization of neural network structure and its training parameters is essential in order to construct narrow prediction intervals with satisfactorily high coverage probability. The second conclusion is that such optimization cannot be carried out through mathematical analysis, mainly due to non-differentiability of the fitness function and its non-monotonous variation. Thus, evolutionary techniques such as Genetic algorithm for locating the optimal configurations and parameters are the ultimate optimization tool.

In this study, a new Genetic algorithm based method for constructing optimal prediction intervals was developed. Its practicality and effectiveness was demonstrated through a case study. Given that the computational burden of the optimization procedure is reasonably inexpensive, it can be applied in every prediction interval construction problem. There is still work to be done for further examination of the proposed method for benchmark datasets and real world case studies, particularly manufacturing enterprises. Also work is in progress to improve quality of the constructed prediction intervals through a comprehensive optimization of all influential factors, including those considered in this study as well as network parameters.

## References

[1] Paliwal, M., Kumar, U.A.: Neural networks and statistical techniques: A review of applications. Expert Systems with Applications 36, 2–17 (2009)

[2] Hannula, M., Huttunen, K., Koskelo, J., Laitinen, T., Leino, T.: Comparison between artificial neural network and multilinear regression models in an evaluation of cognitive workload in a flight simulator. Computers in Biology and Medicine 38, 1163–1170 (2008)

[3] Hwang, J.T.G., Ding, A.A.: Prediction Intervals for Artificial Neural Networks. Journal of the American Statistical Association 92, 748–757 (1997)

[4] de Veaux, R.D., Schumi, J., Jason, S., Ungar, L.H.: Prediction Intervals for Neural Networks via Nonlinear Regression. Technometrics 40, 273–282 (1998)

[5] Nix, D.A., Weigend, A.S.: Estimating the mean and variance of the target probability distribution. In: IEEE International Conference on Neural Networks (1994)

[6] Papadopoulos, G., Edwards, P.J., Murray, A.F.: Confidence estimation methods for neural networks: a practical comparison. IEEE Transactions on Neural Networks 12, 1278–1287 (2001)

[7] Khosravi, A., Nahavandi, S., Creighton, D.: Constructing Prediction Intervals for Neural Network Metamodels of Complex Systems. In: International Joint Conference on Neural Networks, IJCNN 2009 (2009)

[8] Khosravi, A., Nahavandi, S., Creighton, D.: A Prediction Interval-Based Approach to Determine Optimal Structures of Neural Network Metamodels. Expert Systems with Applications (August 2009) (in Press) (accepted Manuscript)

[9] Seber, G.A.F., Wild, C.J.: Nonlinear regression. Wiley, New York (1989)

[10] Lu, T., Viljanen, M.: Prediction of indoor temperature and relative humidity using neural network models: model comparison. Neural Computing & Applications 18, 345–357 (2009)

[11] Gen, M., Cheng, R.: Genetic algorithms and engineering design. Wiley, New York (1997)

[12] Haupt, R.L., Haupt, S.E.: Practical genetic algorithms. Wiley, New York (1998)

[13] Vonk, E., Jain, L.C., Johnson, R.P.: Automatic generation of neural network architecture using evolutionary computation. World Scientific, Singapore (1997)

[14] Bornholdt, S., Graudenz, D.: General asymmetric neural networks and structure design by genetic algorithms. Neural Networks 5, 327–334 (1992)

[15] Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, Irvine, CA, University of California, School of Information and Computer Science (2009), http://www.ics.uci.edu/~mlearn/MLRepository.html

# Involving New Local Search in Hybrid Genetic Algorithm for Feature Selection

Md. Monirul Kabir[1], Md. Shahjahan[2], and Kazuyuki Murase[1]

[1] Department of Human and Artificial Intelligence Systems
University of Fukui, Bunkyo 3-9-1, Fukui 910-8507, Japan
{kabir,murase}@synapse.his.fukui-u.ac.jp
[2] Department of Electrical and Electronic Engineering
Khulna University of Engineering and Technology, Khulna 9203, Bangladesh
jahan@mail.kuet.ac.bd

**Abstract.** This paper presents a new hybrid genetic algorithm (HGA) for feature selection (FS) called as HGAFS. HGAFS incorporates a new local search operation that is devised and embedded in HGA to fine-tune the search in FS. The proposed local search operation works on basis of the distinct and informative nature of input features that is computed by their correlation information. The aim of using correlation information is to encourage the local search strategy for selecting less correlated (distinct) features. Such an encouragement reduces the redundancy of information in the generated subset of salient features. We have tested our methods on several real-world datasets and have compared the performances with the results of other existing algorithms. It is found that HGAFS produces consistently better performances.

**Keywords:** Feature selection, local improvement, correlation information, fitness value.

## 1 Introduction

Feature selection (FS) is a task of reducing the spurious features from the original feature set of a given dataset. Such reduction process ultimately provides the better classification performance in the pattern recognition field and generates a subset of reduced number of salient features. It is known that FS is basically a search process. During searching the spurious features, the approach that depends on the classifier performance as the evaluation function in every FS steps, is called as the wrapper approach [1] while the approach neglecting such evaluation function, is called as filter approach [2]. Furthermore, depending on the selection strategy, searching can be categorized into two ways: forward search [3], and backward search [4]. Apart from these, there are also some other techniques in FS, which are: ant colony optimization (ACO) based search [5] [6], Tabu search, Simulated annealing, and so on.

Genetic search is a recent development guided by genetic algorithm (GA). The GA is biologically inspired and has many mechanisms mimicking natural evolution [7]. It has a lot of prospects in science and engineering optimization or in search problems. Furthermore, GA can be applicable to FS while the problem has an exponential search

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 150–158, 2009.

space. Though GA or, simple GA (SGA) works well in an exponential search space, it suffers by some difficulties such as: inferior solutions caused by premature convergence and poor ability of a fine-tuning near local optimum points [8] [9]. However, to get the better solutions in SGAs, integrating the domain-specific knowledge into SGA, called hybridizing SGA is necessary.

In solving the FS task, there are some hybrid GAs (HGAs) (e.g., [8], [9]) where different types of strategies have been introduced in their local search operations. In [8], the incorporated strategy *ripple*(*r*) operation in which 2*r*-1 times operations are necessary to add the significant features or to delete the insignificant features in (or, from) the selected subset. The computation of significant features here is based on the trained classifier which shows the expensive computational cost. On the other hand, measurement of mutual information (MI) between each pair of features is the main adopted technique in [9] which is also suffered by expensive computation.

As an alternative, this paper proposes a new local search operation in HGA for FS, called as HGAFS that is based on the observation of feature space. In this regard, computation of correlation information is performed to find the relationships between features so that HGAFS can select the distinct and informative features for the pattern classification. The goal of using correlation information is to encourage the search strategy for selecting less correlated (distinct) features. Such encouragement ultimately reduces the redundancy of information in the generated subsets.

A restricted random scheme is also proposed in HGAFS to decide the number of 1-bits (i.e., number of selected features) in individual chromosome of the population set. Such scheme encourages deciding the number of 1-bits in a reduced form. In FS task, it is reasonable in the sense that reduced number of 1-bits ultimately decides the reduced size of subsets. Finally, it can be said that the proposed new local search strategy provides the faster convergence and has an ability to generate reduced subsets of salient features by using its fine-tuning search capability.

The rest of this paper is organized as follows. In Section 2, details of proposed HGAFS are discussed. Experimental results and comparison to other algorithms are reported in Section 3. A short conclusion with few remarks is given in Section 4.

## 2 Proposed Hybrid GAs for Feature Selection

In this paper, the proposed HGAFS consists of two new techniques which are: (a) a random scheme for deciding the number of 1-bits in the chromosome distributions, (b) a new local search operation that improves the quality of newly generated *offsprings*. The above two contributions put impact positively on the final outcomes of HGAFS. Now, the pseudo-code of the local search operation proposed in HGAFS, which can be applicable for a single *offspring* at a time, is outlined as follows,

```
local-improvement(offsp) /* offsp: a offspring */
{
  put features of 1-bits in offsp into $X$;
  compare $X$ with $D$ and $S$ groups; /* $D$: dissimilar group ; $S$: similar group */
  divide $X$ into $X_d$ and $X_s$ accordingly;
  switch{
      $case\ |X_d| < \delta\ and\ |X_s| > \xi : add(\delta - |X_d|); rem(|X_s| - \xi)$
```

```
            case $|X_d| > \delta$ and $|X_s| < \xi$ : $rem(|X_d| - \delta); add(\xi - |X_s|)$
            /* for better understanding see Section 2.2 */
        }
        set 1-bits of $X_d$ and $X_s$ in $X$ accordingly;
        set features of 1-bits in $X$ to offsp;
        set rest of bits in offsp to be 0;
    }
```

In HGAFS, there are some fundamental steps which can be explained as follows,

Step 1) Initialize a feature set *N* of *n* features, a subset *K* of *k* salient features, and a population set *P* of *c* chromosomes. Encode the each string of the chromosome set by binary digits representing the value 1 and 0. The value 1 and 0 represent a feature selected and not selected, respectively. Decide the number of 1-bits in each *c* according to the value of *k* which can be determined by following Section 2.1.

Step 2) Measure the fitness value of chromosome *c* in *P* sequentially using feed-forward three layered neural network (NN) training classifier which can be expressed as,

$$\gamma(Chrom_c) = 100 * (1 - TER) \tag{1}$$

Here, *TER* refers to the testing error rate of the NN on the testing dataset.

Step 3) Perform the conventional 1-point crossover operation [11] by using the conventional rank-based selection procedure [11]. In this case, follow the crossover probability which is defined by user previously.

Step 4) Perform mutation operation according to the conventional scheme [11] over the whole chromosome set in *P* by following the earlier user defined mutation probability.

Step 5) Perform the local-improvement operation upon one generated *offspring* according to the proposed strategy described in Section 2.2. Repeat the same operation until all generated *offsprings* are covered.

Step 6) Replace the chromosomes of lowest rank order in *P* by the new local improved *offsprings.*

Step 7) Check the current generation whether it is equal to the predefined total number of generation *T*, then continue. Otherwise, Go to Step 2).

Step 8) In order to locate the best chromosome that signifies the desired subset of salient features, follow the same procedure that mentioned in [13]. In this case, we select the best chromosome of each generation which is compared with the best chromosome of the previous generation.

HGAFS uses the constructive NN training classifier to compute the fitness of the individual chromosomes. Constructive strategy tries to adjust the suitable number hidden neurons in the hidden layer during training that enhances the classification capability of NN as well. The details description of such strategy can be found in [10]. The following section gives more details about the different components of our proposed algorithm.

### 2.1 Determination of Subset Size

In HGAFS, deciding the size of salient feature subset fully depends upon the number of 1-bits ($k$) in the final best chromosome. It should be noted that the value of $k$ in each chromosome once is decided must be fixed up to the final state of FS process. However, if the value of $k$ in each chromosome is too high or, too low then the fitness value may degrade. Thus, by considering the both issues we propose a modified scheme from [13], called *restricted random scheme*. The aim of such scheme is to maintain the value of $k$ in a reasonable range while designing the chromosome set which can be described by two ways:

Firstly, the probabilistic value of $k$ in a bounded region can be defined as,

$$P(k) = \frac{n-k}{\sum_{i=1}^{l}(n-i)} \qquad where,\ 2 \le k \le \psi\ and\ l \in n-k \tag{2}$$

Here, $\psi$ is define by $\varepsilon$ of $n$. Specifically, $\varepsilon$ is a user specified parameter and its value is set here up to [0.15, 0.4] depending on the number of $n$ of different datasets. The reason is that, if $\psi \approx n$ then the search space for finding the salient feature subset becomes larger which may cause the high computational cost as well as the ineffective subsets.

Secondly, arrange all the possible values of $P(k)$ to the conventional "roulette-wheel selection" scheme [11] to achieve the value of $k$ consistently.

However, in HGAFS, generating the subset is actually maintained by a predetermined range in between 2 to 12 for its size. On basis of this assumption, the value of $\varepsilon$ is determined.

### 2.2 Local Improvement Operation

In HGAFS, the local improvement operation requires the computation of correlation information of features to find the relationships between features. Therefore, HGAFS can detect the distinct and informative features easily. The following four steps describe the proposed local improvement operation.

i) Measure the correlation (degree of relationships) between different features of a given training set using the well-known *Pearson product-moment correlation coefficient* computation. The correlation coefficient $r_{ij}$ between two features $i$ and $j$ is,

$$r_{ij} = \frac{\sum_{p} (x_i - \bar{x}_i)(x_j - \bar{x}_j)}{\sqrt{\sum_{p} (x_i - \bar{x}_i)^2}\sqrt{\sum_{p} (x_j - \bar{x}_j)^2}} \tag{3}$$

where $x_i$ and $x_j$ are the value of features $i$ and $j$, respectively. The variables $\bar{x}_i$ and $\bar{x}_j$ represent the mean values of $x_i$ and $x_j$, averaged over $p$ examples. After computing $r_{ij}$ for all possible combinations of features, HGAFS attempts to compute the correlation of each feature $i$ as,

$$Cor_i = \frac{\sum_{j=1}^{n} r_{ij}}{n-1} \qquad if \quad i \neq j \tag{4}$$

Thereafter, all features *n* are arranged by ascending order according to their correlation values.

ii) HGAFS then creates two groups. One group contains the first *n/2* features, called as *dissimilar* (*D*) *group*, while the other group contains the remaining *n*/2 features, called as *similar* (*S*) *group*. Now, the first feature of both groups is the least correlated (most distinct) among the other features of each individual group.

iii) Perform the local operation upon the newly generated *offspring*. During this operation, distinguish the number of 1-bit in one set *X*. Then, compare each element of *X* with the element of *D* and *S*. Thus, two subsets $X_d$ and $X_s$ are ultimately formed where those contains the features of *D* and *S*, respectively.

iv) Decide that, $X_d$ and $X_s$ always keep a number of features, say, $\delta$ and $\xi$, respectively. Here, $\delta$ is equal to $\mu$ of *k* whereas $\xi$ is to be (1-$\mu$) of *k*. For this, it is necessary to adjust the above quantities in every step. Since, our motivation is to provide more distinct features to the *offspring* set, therefore, $\mu$ here is set to 0.65. However, compare the current $|X_d|$ and $|X_d|$ with $\delta$ and $\xi$, respectively to be made the following two decisions,

(a) $if \ |X_d| < \delta \quad add(\delta - |X_d|); \quad if \ |X_s| > \xi \quad rem(|X_s| - \xi)$

(b) $if \ |X_d| > \delta \quad rem(|X_d| - \delta); \quad if \ |X_s| < \xi \quad add(\xi - |X_s|)$

Here, *add* and *rem* operations are performed according to the distinctness of features from *D* and *S* groups. Specifically, *add*() indicates to add more distinct features comparing to the current ones in $X_d$ or $X_s$ whereas *rem*() specifies to be removed more similar (or, less distinct) features comparing to the present ones in $X_d$ or $X_s$.

## 3 Experimental Setup

HGAFS was applied to four real-world benchmark datasets to evaluate its performance. The datasets are: breast cancer (BCR), glass (GLS), vehicle (VCL), and sonar (SNR) and the details description of these datasets can be found in [14]. The characteristics of the datasets and their partitions are shown in Table 1. Each experiment was carried out 20 times and the presented results are the average of these 20 runs. The performance of HGAFS was evaluated in terms of the number of selected features ($n_s$) as well as classification accuracy (*CA*). All experiments were done in Pentium-core 2 duo, 2.66 GHz personal computer.

In this study, we used a number of user specified parameters and their values are decided in some certain ranges. For example, (a) population size=[20,40], (b) crossover probability=0.06, (c) mutation probability=[0.03,0.05], and (d) generation=[20,40]. There are also some other parameters used in NN training while the string of each chromosomes is evaluated. The initial connection weights for an NN were randomly set to [-1.0, 1.0]. The learning rate and momentum term were set to [0.1, 0.2] and [0.5, 0.9], respectively. The number of partial training epochs of NN was chosen between [10, 70]. The NN training was conducted by the well-known BP algorithm [12]. We conducted one additional set of experiments to investigate the performance of the original all features using constructive NN training classifier.

In course of measuring the fitness value of individual chromosomes, the total examples of the respective dataset were partitioned into three sets. The first 50% examples was selected as a training set to train the NN, the second 25% examples as the validation set to check the condition during training, and the last 25% as the testing set to test the NN.

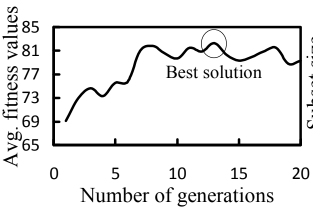


Fig. 1. Generation process of glass dataset for a single run

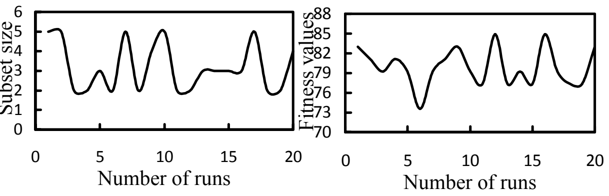


Fig. 2. Variation between subset sizes and fitness values in different runs

### 3.1 Experimental Results

Table 2 shows the average results of using all features and of selected features ($n_s$) by HGAFS. The classification accuracy (*CA*) in the table refers to the percentage of correct classification by the trained NN on the testing set. It is seen that a small number of features from the original feature set was selected by HGAFS. For instance, in case of sonar dataset, HGAFS selected 5.35 features on average from a set of 60 features. This indicates that HGAFS could find a reduced number of salient features. The positive effect of a small number of features can be seen when we look the *CA*. For example, the vehicle dataset, the *CA* of all features was 58.77%, when it was 75.79% with 4.40 features. HGAFS also exhibits good results for other datasets.

Furthermore, the use of $n_s$ causes a small standard deviation (*SD*) as presented in the Table 2 for each entry. The low *SDs* refer to the robustness of HGAFS which is indeed the consistency of an algorithm under different initial conditions.

In order to observe how the generation process of HGAFS progresses, Fig. 1 shows the whole scenery of glass problem for a single run. It is seen that average fitness value of the population is being varied while the generation increases. The circle indicates that the maximum average fitness was achieved at that point. The complete information of that point is: the generation number is 13, subset size is 3, and the *CA* of that subset is 83.02%. In contrast, Fig. 2 exhibits the variation curves between the subset size and *CAs* in total 20 runs. Thus, it can be assumed that the performance of a subset is roughly dependent on its size.

### 3.2 Comparison with Other Works

In this context, the obtained results of HGAFS on four datasets are compared with the results of different FS algorithms. Three well-known algorithms HGAFS$_H$[9], GPFS[13], and ICFS[3] are chosen for comparison. The results are summarized in

Table 3. Since, the different algorithms were evaluated in different experimental setups; therefore, we cannot compare the results of HGAFS completely with other algorithms until the all experiments are performed in the same experimental setup.

**Table 1.** Characteristics of datasets

| Datasets | Features | Classes | Examples | Partition sets | | |
|---|---|---|---|---|---|---|
| | | | | Training | Validation | Testing |
| BCR | 9 | 2 | 699 | 349 | 175 | 175 |
| GLS | 9 | 6 | 214 | 108 | 53 | 53 |
| VCL | 18 | 4 | 846 | 424 | 211 | 211 |
| SNR | 60 | 2 | 208 | 104 | 52 | 52 |

**Table 2.** Average results of BCR, GLS, VCL, and SNR datasets. SD refers standard deviation

| Datasets | Avg. results with all features | | Avg. results with selected features | | | |
|---|---|---|---|---|---|---|
| | CA (%) | SD | No. of feature | SD | CA (%) | SD |
| BCR | 97.60 | 0.002 | 3.25 | 1.17 | 98.55 | 0.006 |
| GLS | 71.51 | 0.046 | 3.45 | 1.07 | 81.04 | 0.021 |
| VCL | 58.77 | 0.152 | 4.40 | 1.15 | 75.79 | 0.009 |
| SNR | 70.87 | 0.092 | 5.35 | 2.21 | 85.87 | 0.037 |

**Table 3.** Comparisons between HGAFS, HGAFS$_H$[9], GPFS[13], and ICFS[3]

| Datasets | | Comparisons | | | |
|---|---|---|---|---|---|
| | | HGAFS | HGAFS$_H$ | GPFS | ICFS |
| BCR | No. of features | 3.25 | -- | 2.23 | 5.00 |
| | Class. acc. (%) | 98.55 | -- | 96.84 | 98.25 |
| GLS | No. of features | 3.45 | 5.00 | -- | 4.50 |
| | Class. acc. (%) | 81.04 | 65.51 | -- | 65.19 |
| VCL | No. of features | 4.40 | 11.00 | 5.37 | -- |
| | Class. acc. (%) | 75.79 | 76.36 | 78.45 | -- |
| SNR | No. of features | 5.35 | 15.00 | 9.45 | -- |
| | Class. acc. (%) | 85.87 | 87.02 | 86.26 | -- |

Table 3 shows the comparisons between HGAFS and other algorithms on basis of average percentage of *CAs* and average number of $n_s$. Now, the comparative studies in between HGAFS and other algorithms for four datasets are stated below.

*Cancer*: The number of $n_s$ in HGAFS is lower than ICFS but comparable with GPFS. In contrast, the overall performance is better than the others.

*Glass*: HGAFS outperforms HGAFS$_H$ and ICFS significantly in every event. HGAFS drastically reduces the original feature set and optimally generates a reduced number of salient feature subset resulting better *CAs*.

*Vehicle*: In terms of number of $n_s$, HGAFS achieved a few number of features comparing to $\text{HGAFS}_\text{H}$ and GPFS while in case of *CA*, it is comparable to those algorithms.

*Sonar*: HGAFS achieved a few number of $n_s$ comparing to $\text{HGAFS}_\text{H}$ and GPFS. But in case of *CA*, it is reasonable or comparable to those algorithms.

## 4 Conclusion

This paper proposes a new local search approach in HGA for FS that is based on the observation of feature space. Computation of correlation information of features was used to recognize the distinct and informative features which were utilized later in the local search operation of HGAFS. Thus, the proposed local search operation helps to reduce the redundancy of information in the generated subset. In contrast, a restricted random scheme was incorporated to decide the number of 1-bits in individual chromosome of a population set which is reasonable in the sense that reduced number of one decides the reduced size of subsets.

In HGAFS, neither the computation of training based classifier nor the computation of mutual information between a pair of features is taken into account for its local search operation. Apart from these both issues HGAFS achieves a faster convergence and fine-tuning in local optimum points during FS.

Extensive experiments have been carried out in this paper to evaluate how well HGAFS performed on different real-world datasets in comparison with other prominent FS algorithms. In almost all except some few cases, HGAFS outperformed the others in terms of the number of selected features and classification performances. The results of the low *SDs* of the *CAs* exhibit the robustness of this algorithm.

Since the focus of this paper is to present the fundamental ideas and technical details of HGAFS, the detailed comparisons with other algorithms using rigorous statistical methods are left as the future work.

## Acknowledgements

Supported by grants to KM from the Japanese Society for Promotion of Sciences, the Yazaki Memorial Foundation for Science and Technology, and the University of Fukui.

## References

1. Hsu, C., Huang, H., Schuschel, D.: The ANNIGMA-wrapper approach to fast feature selection for neural nets. IEEE Trans. on Syst., Man, and Cybern.-Part B: Cybern. 32(2), 207–212 (2002)
2. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: 17th International Conference on Machine Learning (2000)
3. Guan, S., Liu, J., Qi, Y.: An incremental approach to contribution-based feature selection. Journal of Intelligence Systems 13(1) (2004)

4. Abe, S.: Modified backward feature selection by cross validation. In: Proceedings of the European Symposium on Artificial Neural Networks, pp. 163–168 (2005)
5. Aghdam, M.H., Aghaee, N.G., Basiri, M.E.: Text feature selection using ant colony optimization. Expert systems with applicantions 36, 6843–6853 (2009)
6. Ke, L., Feng, Z., Ren, Z.: An efficient ant colony optimization approach to attribute reduction in rough set theory. Pattern Recognition Letters 29, 1351–1357 (2008)
7. Holland, J.: Adaptation in Nature and Artificial Systems. MIT Press, Cambridge (1992)
8. Oh, I.-S., Lee, J.S., Moon, B.: Hybrid Genetic Algorithms for Feature Selection. IEEE Trans. on Pattern Analysis and Machine Intelligence 26(11), 1424–1437 (2004)
9. Huang, J., Cai, Y., Xu, X.: A hybrid genetic algorithm for feature selection wrapper based on mutual information. Pattern Recognition Letters 28, 1825–1844 (2007)
10. Kwok, T.Y., Yeung, D.Y.: Objective functions for training new hidden units in constructive neural networks. IEEE Trans. Neural Network 8(5), 1131–1148 (1997)
11. Goldberg, D.E.: Genetic Algorithms in search, optimization and machine learning (2004)
12. Rumelhart, D.E., McClelland, J.: Parallel distributed processing. MIT Press, Cambridge (1986)
13. Muni, D.P., Pal, N.R., Das, J.: Genetic Programming for Simultaneous Feature Selection and Classifier Design. IEEE Trans. on Systems, Man, and Cybern.-Part B: Cybern. 36(1) (2006)
14. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases. Dept. of Information and Computer Sciences, University of California, Irvine (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html

# Pareto Optimal Based Evolutionary Approach for Solving Multi-Objective Facility Layout Problem

Kazi Shah Nawaz Ripon, Kyrre Glette, Omid Mirmotahari,
Mats Høvin, and Jim Tørresen

Deptartment of Informatics, University of Oslo, Norway
{ksripon,kyrrehg,omidmi,matsh,jimtoer}@ifi.uio.no

**Abstract.** Over the years, various evolutionary approaches have been proposed in efforts to solve the facility layout problem (FLP). Unfortunately, most of these approaches are limited to a single objective only, and often fail to meet the requirements for real-world applications. To date, there are only a few multi-objective FLP approaches have been proposed. However, they are implemented using weighted sum method and inherit the customary problems of this method. In this paper, we propose an evolutionary approach for solving multi-objective FLP using multi-objective genetic algorithm that presents the layout as a set of Pareto optimal solutions optimizing both quantitative and qualitative objective simultaneously. Experimental results obtained with the proposed algorithm on test problems taken from the literature are promising.

**Keywords:** Multi-objective facility layout problem, Pareto optimal solutions, Quantitative objective, Qualitative objective.

## 1 Introduction

FLPs deal with arranging a given number of facilities (departments) on the factory floor of a manufacturing system to meet one or more objectives. These objectives may include minimizing the total cost of transporting materials (material handling costs) or maximizing adjacency requirement between the facilities. In essence, FLP can be considered as a searching or optimization problem, where the goal is to find the best possible layout. Traditionally FLP has been presented as a Quadratic Assignment problem (QAP) to find the best assignment of $n$ facilities to $m$ locations, where the number of facilities and locations must be equal. It is well known that QAP is NP-complete category due to the combinatorial function involved and cannot be solved for large layout problems. Despite its popularity, QAP is a difficult problem to solve using traditional optimal algorithms [1].

During the last decades many different methods have been developed to solve FLP with genetic algorithms (GAs). However, most of the researches in this field are generally concerned with a single objective, either qualitative or quantitative feature of the layout. By contrast, practical layout problems are multi-objective by nature and they require the decision makers to consider a number of criteria involving both quantitative and qualitative objectives before arriving at any conclusion. A solution that is optimal with respect to a given criterion might be a poor candidate for some other

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 159–168, 2009.

criteria. Hence, the trade-offs involved in considering several different criteria provide useful insights for the decision makers. Surprisingly, the research in this important field has been scarce when compared to the research in single criterion.

Although dealing with multiple objectives has received attention over the last few years [2,3,4,5], to our knowledge these approaches are still considered limited, and mostly dominated by the unrealistic aggregation of preferences method, popularly known as weighted sum method. In this method, multiple objectives are combined into a single scalar objective using weighted coefficients. However, there are several disadvantages of this technique [6]. First, as the relative weights of the objectives are not exactly known in advance and cannot be pre-determined by the users, the objective function that has the largest variance value may dominate the multi-objective evaluation. As a result, inferior non-dominated solutions with poor diversity will be produced. Second, the user always has to specify the weight values for functions and sometimes this will not have any relationship with the importance of the objectives. Third, a single solution is obtained at one time. If we are interested in obtaining a set of feasible solutions, it has to be run several times. This also, is not a warranty that the solutions obtained in different runs are different. More importantly, since the selection of objective weights is critical in designing layout having multiple objectives, the objective weights therefore play an important role in the design process. In practice, it is selected randomly by the layout designer based on his/her past experience that restricts the designing process completely designer dependent and thus the layout varies from designer to designer. To overcome such difficulties, Pareto-based evolutionary optimization has become an alternative to classical weighted sum method. Goldberg [7] first proposed this approach and it explicitly uses Pareto dominance to determine the reproduction probability of each individual.

This paper presents a multi-objective evolutionary approach for FLP to find a set of Pareto optimal layout solutions optimizing both quantitative and qualitative objectives. In this work, we have used the Non-dominated Sorting Genetic Algorithm 2 (NSGA 2) proposed by Deb et. al. [8]. The goal of this proposed multi-objective FLP approach is to find as many different potential layouts as possible, each of which is near optimal and is not dominated by consideration of a particular objective. In an attempt to address multiple objectives simultaneously in this work, we apply material handling costs and closeness relationship among various departments as quantitative and qualitative objective respectively.

The paper is organized as follows. Section 2 explains the concept of multi-objective optimization and Pareto-optimal solution. Section 3 describes the related works on FLP. Section 4 justifies the necessities of multi-objective evolutionary FLP algorithm. The implementation of multi-objective FLP algorithm is presented in Section 5. Experimental results are presented and analyzed in Section 6, followed by the conclusions in the final section.

## 2 Multi-Objective Optimization and Pareto Optimal Solution

In the world around us, there are few problems concerned with a single value or objective. Instead, most problems involve multiple objectives and constraints that often conflict with each other. For such multi-objective optimization problems (MOOP), these conflicts have to be met or optimized before any adequate solution is reached. However, it is rarely the case that a single solution can simultaneously satisfy all the existing objectives. Therefore, when dealing with MOOPs, we normally look for the trade-offs rather than a single solution. In order to generate these trade-off solutions, an old notion of optimality called Pareto-optimum [9] is normally adopted. In a MOOP, a solution $x^{(1)}$ is said to dominate the other solution $x^{(2)}$, if both the following conditions are true: (1) The solution $x^{(1)}$ is no worse than $x^{(2)}$ in all objectives, and (2) the solution $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective.

For a given finite set of solutions, we need to perform pair-wise comparisons to find out which solutions dominate and which are dominated by each other. From these comparisons, we can find a subset of the finite set of solutions such that, any two solutions of which do not dominate each other and all the other solutions of the finite set are dominated by one or more members of this subset. This subset is called the non-dominated set for the given set of solutions. Unfortunately the use of this concept almost always gives not a single solution but a set of solutions, which is called the Pareto-optimal set.

## 3 Related Works

The FLP has been extensively studied over the last few decades and a wide variety of approaches have been proposed. Traditionally, these approaches have been divided into two categories [10]: optimal and suboptimal. The optimal methods such as the branch-and-bound and cutting plane algorithm have been successfully applied to FLP when the number of facilities is less than 15. However, as the number of departments is larger than 15, QAP has been validated to be an NP-complete problem and its computational time is exponentially increased [1].

FLP is one of the truly difficult ill-structured, multi-criteria and combinatorial optimization problems. In recent years, a lot of sub-optimal and intelligent techniques have been developed to cope with this type of problems. Most of these approaches belong to heuristic ones such as simulated annealing [11], tabu search [12], and GAs [13, 14]. During the past three decades, numerous heuristic methods have been developed to obtain some good, rather than optimal, solutions for layout problems. However, Generally speaking, GAs outperform other heuristic methods [1]. GAs have been widely implemented to solve combinatorial optimization problems and are considered to be a robust approach by accompanying them with artificial intelligence [9]. Comprehensive surveys of the different approaches to FLPs are found in [10,15].

## 4 Importance of Multi-Objective FLP

Historically, FLPs have been solved only for one goal, either quantitative or qualitative aspect of the layout. Quantitative approaches involve primarily the minimization of material handling costs between various departments. More specifically, it tries to minimize the sum of distance among all facilities multiplied by the corresponding flows. Qualitative approaches used the closeness rating scores to indicate the desired relative “closeness” requirement for two departments to be next to each other. Here the goal is to maximize the total closeness ratings among all departments. The closeness ratings are; A (absolutely necessary), E (essentially important), I (important), O (ordinary), U (un-important) and X (undesirable), to indicate the respective degrees of necessity that any two given departments be located close together.

In real-world FLPs, it is often necessary to optimize both quantitative and qualitative criteria simultaneously. In general, the minimization of the total material handling costs is often used as the optimization criterion in FLP. However, the closeness rating, hazardous movement or safety, and the like are also the important criteria in FLP. Many researchers have questioned the appropriateness of selecting a single-criterion objective to solve FLP because qualitative and quantitative approaches each have advantages and disadvantages [4]. The major limitations on quantitative approaches are that they consider only relationships that can be quantified and do not consider any qualitative factors. The shortcoming of qualitative approaches is their strong assumption that all qualitative factors can be aggregated into one criterion. In essence, FLPs fall into the category of a MOOP.

Accordingly, it is desirable to generate many near-optimal layouts considering multiple objectives according to the requirements of the production order or customer demand. Then, the production manager can selectively choose the most demanding one among all of the generated layouts for specific order or customer demand. On the other hand, if multiple objectives conflict with each other, then the production manager does not need to omit any required objective. Based on the principle of multi-objective optimization, obtaining an optimal solution that satisfies all of the objectives is almost impossible. However, it is desirable to obtain as many different Pareto-optimal layout solutions as possible, which should be non-dominated, converged to, and diverse along the Pareto-optimal front with respect to these multiple criteria.

# 5 The Proposed Pareto Optimal Based Approach

## 5.1 Chromosome Representation

In this study, a form of direct representation for strings is used. The solution is represented as a string of integers of length *n*, where *n* is the number of facilities. The integers denote the facilities and their positions in the string denote the positions of the facilities in the layout. For example, the following assignment of the 8/8 problem

7 6 5 3
8 4 1 2

will be represented by the solution string 7 6 5 3 8 4 1 2.

## 5.2 Fitness Evaluation and Selection Scheme

As discussed early, there are two approaches in facility layout algorithms: minimize the distance-based objective function value (quantitative approach), and maximize the adjacency-based objective function (qualitative approach). In this work, we separately utilized both of these as objectives. The first fitness function, total material handling cost, is based on quantitative model. This function is subject to minimization, and measured as

$$Z_x = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} A_{ijkl} X_{ij} X_{kl} \tag{1}$$

subject to

$$\sum_{i=1}^{n} X_{ij} = 1; j = 1,...n \quad , \quad \sum_{j=1}^{n} X_{ij} = 1; i = 1,...n \quad , \text{and} \quad X_{ij} = 0 or 1$$

providing that

$$X_{ij} = \begin{cases} 1, if \; facility\, i\, is\, assigned\, to\, location\, j \\ 0; otherwise \end{cases}$$

$$X_{kl} = \begin{cases} 1, if \; facility\, k\, is\, assigned\, to\, location\, l \\ 0; otherwise \end{cases}$$

$$A_{ijkl} = \begin{cases} f_{ik} d_{jl} & if\; i \neq k\, or\; j \neq l \\ f_{ii} d_{jj} + c_{ij} & if\; i = k\, and\; j = l \end{cases}$$

where

$c_{ij}$ = cost of assigning facility *i* to location *j*

$d_{jl}$ = distance from location *j* to location *l* (rectilinear distance)

$f_{ik}$ = material flow from facility *i* to facility *k*.

The second fitness function, the closeness rating score, is based on qualitative model. This function is subject to maximization, and measured as

$$Z_y = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} W_{ijkl} X_{ij} X_{kl} \tag{2}$$

subject to

$$\sum_{i=1}^{n} X_{ij} = 1; j = 1,...n \quad , \quad \sum_{j=1}^{n} X_{ij} = 1; i = 1,...n \quad , \text{and} \quad X_{ij} = 0 or 1$$

where

$$W_{ijkl} = \begin{cases} r_{ik}, if \; locations\; j\, and\, l\, are\, neighbors \\ 0, otherwise \end{cases}$$

$r_{ik}$ = closeness ranking value when facility *i* and *k* are neighbors with common boundary.

The numerical values used for the closeness ranking values are: A=6, E=5, I=4, O=3, U=2, and X=1. After performing all these operations, a non-dominated sorting strategy employing the crowding-distance assignment [8] is performed to achieve the elitisms for the next generation.

### 5.3 Crossover and Mutation Operation

The chromosome representation in this multi-objective FLP is different from that of the conventional GA. As a result, the direct application of the traditional genetic operators may create an illegal solution. Hence, some problem-specific genetic operators are required. We follow the concept described by Suresh et. al [16] for crossover operation. It may be noted that this crossover maintains partial structure of the parents to a large extent than by the existing crossover operations such as PMX, OX and CX [16]. Mutation provides and maintains diversity in a population. In our approach, two genes are picked up randomly and then they exchange their positions. As a result, the resultant chromosomes are legal and no repair is required.

## 6 Experimental Results and Discussion

Until now, almost all FLP algorithms try to optimize single criteria only (mainly minimizing the material handling cost). Though few multi-objective FLP approaches can be found in the literature, all of these existing approaches employed weighted sum approach. And to our knowledge, there are no published paper for the Pareto optimal solution. Therefore, to evaluate our proposed algorithm, first we compared the material handling costs obtained by our approach with the existing single objective approaches to justify its capability to optimize this cost. Then we demonstrated its performance as a multi-objective evolutionary FLP algorithm by optimizing material handling cost and closeness rating score. Note that, for both cases we have used the same results achieved by our approach.

**Table 1.** Comparison with existing algorithms

| **Problem (naug n)** | **Lower Bound** | **Best Known** | **H63** | **H63-66** | **CRAFT** | **Biased Sampling** | **FLAC** | **DISCON** | **FATE** | **TAA** | **GESA** | **HU & Wang** | **Proposed Approach** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 41 | 43 | 43<br>(44.2) | 43<br>(44.2) | 43<br>(44.2) | 43<br>(44.2) | 43<br>(43) | 43<br>(47.5) | 50.6<br>(50.6) | 43<br>(43) | 43<br>(43) | 43<br>(43) | 43<br>(43) |
| 8 | 91 | 107 | 109<br>(114.4) | 107<br>(110.2) | 107<br>(113.4) | 107<br>(107) | 107<br>(107) | 107<br>(118.8) | 126.7<br>(126.7) | 116<br>(116) | 107<br>(107) | 107<br>(107.8) | 107<br>(107.75) |
| 12 | 243 | 289 | 301<br>(317.4) | 304<br>(310.2) | 289<br>(296.2) | 289<br>(293) | 289<br>(289) | 295<br>(322.2) | 326.2<br>(326.2) | 314<br>(314) | 289<br>(289.36) | 289<br>(290.6) | **287**<br>(292.57) |
| 15 | 479 | 575 | 617<br>(632.6) | 578<br>(600.2) | 583<br>(600) | 575<br>(580.2) | 585<br>(585) | 597<br>(630.8) | 660.8<br>(660.8) | 596<br>(596) | 575<br>(575.18) | 575<br>(576.4) | 575<br>(677.01) |
| 20 | 1014 | 1285 | 1384<br>(1400.4) | 1319<br>(1345) | 1324<br>(1339) | 1304<br>(1313) | 1303<br>(1303) | 1376<br>(1416.4) | 1436.3<br>(1436.3) | 1414<br>(1414) | 1285<br>(1287.38) | 1285<br>(1290.5) | 1286<br>(1290.6) |
| 30 | 2238 | 3062 | 3244<br>(3267.2) | 3161<br>(3206.8) | 3148<br>(3189.6) | 3093<br>(3189.6) | 3079<br>(3079) | 3330<br>(3436.4) | 3390.6<br>(3390.6) | 3326<br>(33269 | 3062<br>(3079.32) | 3064<br>(3075.1) | 3062<br>(3081.02) |

To evaluate the proposed multi-objective FLP approach, we run the algorithm on various benchmark data. The 6, 8, 12, 15, 20 and 30 facility problem (*naug6*, *naug8*, *naug12*, *naug15*, *naug20*, *naug30*) formulated by Naugent et al. [17], the 6 and 8 facility problem proposed by Dutta and Sahu (*ds6*, *ds8*) [18], the 9 facility problem (*ct9*) used in [19], the 6 and 8 facility problem (*singh6*, *singh8*) proposed in [2], and 42, 72 and 100 facility problem (*sko42*, *sko72*, *wil100*) available in the web [20] are used here. As discussed above, very few benchmark problems are available for multi-objective FLP, particularly, in the case of closeness rating score. Thus, the authors have themselves created their own data sets for closeness rating score. The experiments are conducted using 50 chromosomes and 40 generations for up to 15 facility problem, where as 100 chromosomes and 80 generations for more than 15 facility problem. The probabilities of crossover and mutation are 0.9 and 0.3 respectively. Each benchmark problem is tested for thirty times with different seeds. Then each of the final generation is combined and a non-dominated sorting is performed to constitute the final non-dominated solutions.

**Table 2.** Comparison of the material handling cost obtained by various evolutionary methods

| **Problem** | ***n*** | **SA** | **TS** | **GAs** | **Proposed Approach** |
|---|---|---|---|---|---|
| naug30 | 30 | 6128 | 6150 | 6202 | 3062 |
| sko42 | 42 | NA | 15866 | 15982 | 15796 |
| sko72 | 72 | NA | 66920 | 67160 | 66034 |
| will100 | 100 | 274022 | NA | 275290 | 273988 |

## 6.1 Single Objective Context

In Table 1, the performance of the proposed approach is compared with some existing algorithm in term of material handling cost. This table is partially cited from [1] and value in the parenthesis represents the average objective function value. From this table, it can be easily found that the proposed FLP algorithm is capable in producing near-optimal values for the test problems. As indicated in the table, the performance of the proposed approach is superior or equivalent to other approaches and match up with most of the best-found solutions; although some other algorithms may achieve this value. In case of *naug12*, it achieves the new best-found material handling cost. Only exception is in the case of *naug20* problem, where it fails to achieve the best-known result by a slight margin. Also, in some cases, the average results obtained by the proposed algorithm are a little higher than that of some existing algorithms. Despite that it should be mentioned that all the compared algorithms are designed for single objective only, and the main goal of our proposed algorithm is to find trade-off solutions for multi-objective FLP, which is very rare in literature. Also, according to the Pareto optimal theory, the final and average value may be influenced by the presence of other objective (closeness rating score). While considering this, the overall performance of the proposed approach is very promising for all the problems.

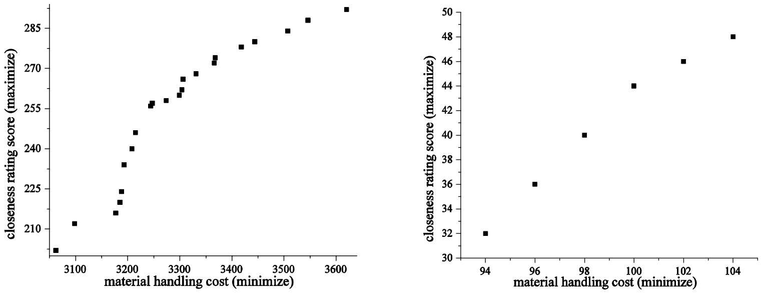
**Table 3.** Results of Test Problems

| Problem | Material Handling Cost | | Closeness rating Score | |
|---|---|---|---|---|
| | Best | Average | Best | Average |
| ds6 | 96 | 96.8 | 48 | 43.40 |
| ds8 | 179 | 209.84 | 82 | 70.3 |
| ct9 | 4818 | 4822.9 | 90 | 74.79 |
| singh6 | 94 | 98.28 | 48 | 40.48 |
| Singh8 | 179 | 199.84 | 82 | 73.1 |
| naug30 | 3062 | 3081.02 | 292 | 254.05 |

We also perform experiments to compare our proposed evolutionary FLP algorithm with some existing evolutionary FLP approaches to justify its efficiency as an evolutionary approach. Table 2 summarizes the material handling cost in comparison with these methods for *naug30*, *sko42*, *sko72*, *will00* problems. The column headings, SA, TS and GAs indicate simulated annealing, tabu search and genetic algorithm based approaches respectively. This table is partially cited from [21]. From the table, it can be found that the proposed approach clearly outperforms the other evolutionary approaches by a significant margin.

## 6.2 Multiple-Objective Context

MOEAs do not try to find one optimal solution but all the trade-off solutions, and deal with two goals - finding a set of solutions as close as possible to the Pareto-optimal front and as diverse as possible. Table 3 shows the performance statistics of the evolutionary multi-objective FLP in the context of material handling cost and closeness rating score. All the results for these problems available in the literature used weighted sum approach for handling multiple objects and present only a single final value. However, by extracting the material handling cost from these results, it can be shown that our proposed algorithm outperforms the best-known results for *ds6* and *ds8* problems. The available best-known results are 98 and 190 respectively. In others cases, the results are the same. On the other hand, the closeness rating score may vary in different cases. This is due to the fact that different authors use different rating scores for the test problems.



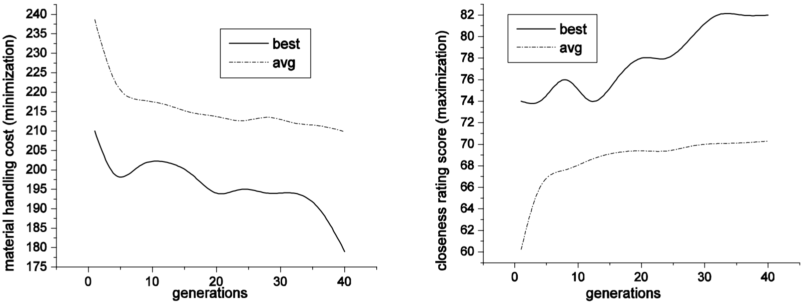
(a) naug30

(b) singh6

**Fig. 1.** Final Pareto-optimal front

To illustrate the convergence and diversity of the solutions, non-dominated solutions of the final generation produced by the proposed algorithm for the test problems *naug30* and *singh6* are presented in Fig. 1. From these figures, it can be observed that the final solutions are well spread and converged. And for this reason it is capable of finding extreme solutions. It is worthwhile to mention that in all cases, most of the solutions of the final population are Pareto optimal. In the figures, the occurrences of the same non-dominated solutions are plotted only once.

Fig. 2 demonstrates the convergence behavior of the proposed methods over generations for *ds8*. These figures also justify that our proposed approach clearly optimizes both of the objectives with generations. From the figures, it can be found that from first generations to last generations, the proposed method is able to optimize both of the material handling cost (minimize) and closeness rating score (maximize).



(a) Material handling cost (b) closeness rating scale

**Fig. 2.** Two objectives over generations of *ds8* problem

To summarize the result, the proposed approach is capable of producing near-optimal and non-dominated (Pareto optimal) solutions, which are also the best-known results in many cases. The simulation results clearly show that our proposed approach is able to find a set of diverse Pareto optimal solutions, which fulfills the two main goals of the multi-objective FLP algorithm.

## 7 Conclusion

FLP has attracted the attention of many researchers because of its practicality and interdisciplinary importance. Although several schemes for solving FLP are available in the literature, very few of them deal with multi-objective approach to optimize both qualitative and quantitative objectives. This is in contrast to real life, where FLP must consider both types of objectives. At present, several multi-objective FLP techniques have been proposed. However, most of them are limited to the weighted sum approach and suffer from a number of problems. In this work, we have presented an evolutionary approach for solving the multi-objective FLP that searches for the Pareto-optimal solutions. The experimental results demonstrate that the proposed approach can produce an overall strong performance for all of the applied benchmark problems related to the material handling cost in the context of single-objective optimization. For the multi-objective optimization, the results show that it is capable in finding a set of Pareto optimal solutions that optimizes both material handling cost and closeness rating score simultaneously throughout the evolutionary process, considering both diversity and convergence of the non dominated solutions.

## References

1. Michael, H.H., Ming-Jaan, W.: Using Genetic Algorithms on Facilities Layout Problems. Int. J. Adv. Manuf. Technol. 23, 301–310 (2004)
2. Singh, S.P., Singh, V.K.: An Improved Heuristic Approach for Multi-Objective Facility Layout Problem. Int. J. Prod. Res. iFirst, 1–24 (2009)

3. Ye, M., Zhou, G.: A Local Genetic Approach to Multi-Objective, Facility Layout Problems with Fixed Aisles. Int. J. Prod. Res. 45(22), 5243–5264 (2007)
4. Chen, C.-W., Sha, D.Y.: Heuristic Approach for Solving the Multi-Objective Facility Layout Problem. Int. J. Prod. Res. 43(21), 4493–4507 (2005)
5. Shouman, M.A., Nawara, G.M., Mohamed, H.E., El Shaer, R.H.: Genetic Algorithm Approach for Solving Multi-Objective Facility Layout Problem. Alexandria Engineering Journal 43(3), 285–295 (2004)
6. Grosan, C., Abraham, A., Tigan, S., Chang, T.G.: How to Solve a Multicriterion Problem for Which Pareto Dominance Relationship Cannot Be Applied? A Case Study from Medicine. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4253, pp. 1128–1135. Springer, Heidelberg (2006)
7. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transaction on Evolutionary Computation 6(2), 182–197 (2002)
9. Pareto, V.: Cours D'Economie Politique, Rouge, Lausanne, Switzerland (1896)
10. Kusiak, A., Heragu, S.S.: The Facility Layout Problem. Eur. J. Oper. Res. 29, 229–251 (1987)
11. Chwif, L., Marcos, R.P.B., Lucas, A.M.: A Solution to The Facility Layout Problem Using Simulated Annealing. Comput. Ind. 36, 125–132 (1998)
12. Chiang, W.C., Kouvelis, P.: An Improved Tabu Search Heuristic for Solving Facility Layout Design Problems. Int. J. Prod. Res. 34(9), 2565–2585 (1996)
13. Kluendran, B., Velappan, S., Balamurugan, I.: Manufacturing Facilities layout Design using Genetic Algorithm. Int. J. Manufacturing and Management 14(3/4), 461–474 (2008)
14. Al-Hakim, L.: On Solving Facility Layout Problems using Genetic Algorithms. Int. J. Prod. Res. 38(11), 2573–2582 (2000)
15. Singh, S.P., Sharma, R.R.K.: A Review of Different Approaches to the Facility Layout Problems. Int. J. Adv. Manuf. Technol. 30, 425–433 (2006)
16. Suresh, G., Vinod, V.V., Sahu, S.: A Genetic Algorithm for Facility Layout. Int. J. Prod. Res. 33(12), 3411–3423 (1995)
17. Nugent, C.E., Vollmann, T.E., Ruml, J.: An Experimental Comparison of Techniques for the Assignment of Facilities to Locations. Operations Research 16(1), 150–173 (1968)
18. Dutta, K.N., Sahu, S.: A Multigoal Heuristic for Facilities Design Problems: MUGHAL. Int. J. Prod. Res. 20(2), 147–154 (1982)
19. Chan, K.C., Tansri, H.: A Study of Genetic Crossover Operations on the Facility Layout Problem. Computers & Industrial Engineering 126(3), 537–550 (1994)
20. http://www.opt.math.tu-graz.ac.at/qaplib/
21. Tavakkoli, M.R., Shayan, E.: Facilities Layout Design by Genetic Algorithms. Computers and Industrial Engineering 35(3/4), 527–530 (1998)

# Swarm Reinforcement Learning Algorithm Based on Particle Swarm Optimization Whose Personal Bests Have Lifespans

Hitoshi Iima and Yasuaki Kuroe

Kyoto Institute of Technology,
Matsugasaki, Sakyo-ku, Kyoto, Japan
{iima,kuroe}@kit.ac.jp

**Abstract.** We recently proposed a swarm reinforcement learning algorithm based on particle swarm optimization (PSO) in order to find optimal policies rapidly. In this algorithm, multiple agents are prepared, and they learn not only by individual learning but also by an update procedure of PSO. In this procedure, state-action values are updated based on the personal best and the global best which are found by the agents so far. In this paper, we direct our attention to a problem that overvaluing personal bests brings inferior learning performance. In order not to update the state-action values based on the overvalued personal best, we propose a swarm reinforcement learning algorithm based on PSO in which the personal best of each agent has a lifespan.

**Keywords:** reinforcement learning, particle swarm optimization, swarm intelligence.

## 1 Introduction

In ordinary reinforcement learning algorithms [1], a single agent learns to achieve a goal through experiencing many episodes. If a learning problem is complicated, it may take much computation time to acquire the optimal policy. Meanwhile, for optimization problems, population-based methods such as genetic algorithms and particle swarm optimization (PSO) [2] have been recognized that they are able to find rapidly global optimal solutions for multi-modal functions with wide solution space. It is expected that by introducing the concept of population-based methods into reinforcement learning algorithms, optimal policies can be found rapidly.

In order to find optimal policies rapidly, we recently proposed reinforcement learning algorithms using multiple agents [3,4], and call them *swarm reinforcement learning algorithms*. In the algorithms, multiple agents are prepared and they all learn concurrently with two learning strategies: individual learning and learning through exchanging information. In the former strategy, each agent learns individually by using a usual reinforcement learning algorithm such as Q-learning [5]. In the latter strategy, the agents exchange their information among

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 169–178, 2009.

them and learn based on the exchanged information every after repeating the individual learning a certain number of times.

Learning methods called multi-agent reinforcement learning have been proposed [6]. Basically, the aim of the multi-agent reinforcement learning methods is to acquire optimal policies in tasks achieved by cooperation or competition among multiple agents. In the methods each agent regards other agents as a part of environments. The concept and objective of the swarm reinforcement learning algorithms are different from those of the multi-agent reinforcement learning methods. Basically, the swarm reinforcement learning algorithms could treat both tasks achieved by a single agent and achieved by cooperation or competition among multiple agents. In the methods, multiple agents are prepared in order to learn in shorter learning time. In [3] and [4], swarm reinforcement learning algorithms for problems of single-agent tasks are proposed.

The performance of the swarm reinforcement learning algorithms highly depends on a method of exchanging the information, which should be appropriately designed. In our former paper [4], we proposed the swarm reinforcement learning algorithm based on PSO (SRL-PSO) in which the update equations of PSO are used for exchanging the information. In this algorithm, Q-values (state-action values) of each agent are evaluated by a certain way every episode in order to select personal bests and the global best. The superior Q-values among the Q-values found by each agent so far are stored as the personal best. Moreover, the superior personal best among all the personal bests is stored as the global best. In learning through exchanging the information, the Q-values of each agent are updated based on its own personal best and the global best.

In this paper, we point out a problem concerning the evaluation of Q-values in SRL-PSO, and propose a method to resolve it. Although the objective of reinforcement learning is to maximize the return, it is almost impossible to evaluate the Q-values by the return whose calculation requires many simulations. In SRL-PSO, the Q-values are evaluated by a value approximate to the return. Therefore, they are not necessarily evaluated correctly and are sometimes overvalued, which is an inevitable problem in SRL-PSO. When such overvalued Q-values become the personal best, it is hard for the agent to find Q-values which are superior to the overvalued personal best. Since the Q-values of the agent continue to be updated based on the overvalued personal best until superior Q-values are found, the learning speed deteriorates.

In order to resolve the above-mentioned problem, this paper proposes SRL-PSO in which the personal best has a lifespan (SRL-PSO-L). In the proposed algorithm, if the personal best of an agent is not replaced after experiencing a certain number of episodes corresponding to a specified lifespan, it comes the end of its life and its Q-values are replaced with the current Q-values of the agent. Thus, the overvalued personal best is replaced, which resolves the problem of SRL-PSO. SRL-PSO-L is applied to a shortest path problem, and its performance is examined through numerical experiments.

The rest of this paper is organized as follows. Section 2 outlines the swarm reinforcement learning method [3,4]. Next, Section 3 proposes SRL-PSO-L, and Section 4 shows its experimental results. Finally, Section 5 concludes the paper.

## 2 Swarm Reinforcement Learning Method

### 2.1 Basic Framework

The swarm reinforcement learning method [3,4] is motivated by population-based methods in optimization problems and its basic framework is as follows. Multiple agents are prepared and they all learn concurrently with two learning strategies: individual learning and learning through exchanging information. In the former strategy, each agent learns individually by using a usual reinforcement learning algorithm. In the latter strategy, the agents exchange their information among them and learn based on the exchanged information every after repeating the individual learning a certain number of times.

Any usual reinforcement learning algorithm can be used for the individual learning, and Q-learning [5] which is a typical reinforcement learning algorithm is used in this paper. In Q-learning, when agent $i$ takes action $a$ in state $s$, the Q-value $Q_i(s,a)$ of the agent $i$ for the state-action is updated by

$$Q_i(s,a) \leftarrow Q_i(s,a) + \alpha \; \{ \; r + \gamma \max_{a^* \in A(s_\mathrm{n})} Q_i(s_\mathrm{n}, a^*) - Q_i(s,a) \; \} \tag{1}$$

where
$\alpha$ : learning-rate parameter,
$r$ : reward,
$\gamma$ : discount-rate parameter,
$s_\mathrm{n}$ : next state,
$A(s)$ : set of allowable actions in state $s$.

The $\varepsilon$-greedy method is used as an action selection method for all the agents. In this method, each agent takes an action randomly with probability $\varepsilon$, and the action whose value is maximum $(\max_{a^*} Q(s,a^*))$ with probability $1-\varepsilon$.

In the learning through exchanging the information, each agent updates its own Q-values by referring to the Q-values which are evaluated to be more useful and superior to those of the other agents for finding rapidly the optimal Q-values. For this purpose, the Q-values of each agent are evaluated for each episode after the individual Q-learning is performed. In ordinary Q-learning with a single agent, Q-values are not necessary to be evaluated. By contrast, in the swarm reinforcement learning method, it is essential to introduce an appropriate criterion to evaluate the Q-values. By the evaluations, superior Q-values can be selected and exchanged among agents, and Q-values of each agent can be updated by using the superior Q-values.

The flow of the swarm reinforcement learning method is as follows.

Step 1. Each of multiple agents updates its own Q-values by performing the individual Q-learning (1) for a specified number of episodes.

Step 2. The Q-values of each agent are evaluated by an appropriate method, which is described in Subsection 2.2.

Step 3. Based on the evaluation of Step 2, superior Q-values are selected and exchanged among agents. The Q-values of each agent are updated by using the superior Q-values.

Step 4. If the termination condition is satisfied, terminate this algorithm. Otherwise, return to Step 1.

In SRL-PSO [4], the update equations of PSO are used for the procedure of Step 3, which is described in Subsection 2.3.

### 2.2 Evaluation of Q-Values

As stated above, in the swarm reinforcement learning method, it is necessary to appropriately evaluate the Q-values of each agent at the end of each episode. Since the objective of reinforcement learning is to maximize the return, it seems to be most suitable to evaluate the Q-values by directly calculating the return. However, it is not practical, because this calculation requires many simulations. Instead, the Q-values are evaluated in such a way that the evaluation results are close approximations of the returns.

Basically, in this method, the Q-values of each agent are evaluated by the sum of rewards which the agent gains during the previous one episode. However, to simply sum up them causes the following problem. Since a Q-value is updated whenever an agent takes an action, Q-values at and shortly after the beginning of the episode are rather different from those at the end of the episode. Even if a new episode begins with the same Q-values at the end of the previous episode, the same actions are not necessarily taken and the same rewards are not necessarily obtained. The rewards obtained by using the Q-values at and shortly after the beginning are considered to have only a little relation with the Q-values at the end. Thus, such rewards are discounted and the discounted results are summed up. Therefore, the evaluated value $E$ for the Q-values is defined as

$$E = \sum_{k=1}^{N} d^{N-k} r_k \tag{2}$$

where $N$ is the number of actions in the episode, $r_k$ is the reward for the $k$-th action, and $d (< 1)$ is the discount parameter. The definition (2) could bring that the larger the evaluated value $E$ for the Q-values of an agent is, the superior the Q-values are.

### 2.3 Information Exchange Method Based on Particle Swarm Optimization

This subsection describes an information exchange method based on PSO in SRL-PSO. PSO [2] is a population-based method which is often used for rapidly finding global optimal solutions in optimization. PSO originates in social behavior, and each agent updates its own candidate solution by utilizing its own personal best and the global best.

Here, PSO is outlined. Consider a problem of determining values of $N$ decision variables $x = (x(1), x(2), \cdots, x(N))$ which maximize an objective function $f(x)$. In PSO, a swarm is prepared and all particles in the swarm are used for solving this problem. Let $J$ be the number of the particles (the swarm size), and $x_j = (x_j(1), x_j(2), \cdots, x_j(N))$ be the decision variable vector (the candidate solution) of particle $j$ $(j = 1, 2, \cdots, J)$. The candidate solution of $j$ is updated as follows:

$$v_j(n) \leftarrow wv_j(n) + c_1r_1\{p_j(n) - x_j(n)\} + c_2r_2\{g(n) - x_j(n)\} \tag{3}$$

$$x_j(n) \leftarrow x_j(n) + v_j(n) \tag{4}$$

$$(j = 1, 2, \cdots, J; n = 1, 2, \cdots, N)$$

where
$v_j = (v_j(1), v_j(2), \cdots, v_j(N))$ : velocity vector of particle $j$,
$w$ : weight parameter called *inertia weight*,
$c_1, c_2$ : weight parameters called *acceleration coefficients*,
$r_1, r_2$ : uniform random numbers in the range from 0 to 1,
$p_j = (p_j(1), p_j(2), \cdots, p_j(N))$ :
best solution found by particle $j$ so far, which is called *personal best*,
$g = (g(1), g(2), \cdots, g(N))$ :
best solution found by all particles so far, which is called *global best*.

The reinforcement learning problems can be considered to be a kind of optimization problems by regarding a candidate solution and an objective function value as Q-values and an evaluated value $E$, respectively. It is expected that optimal Q-values can be found rapidly by applying the procedure of updating the candidate solution in PSO to the update scheme of Q-value.

In this information exchange method, the personal best of each agent and the global best are determined by comparing evaluated values and are stored. Let the personal best $P_i$ be the best Q-values found by the agent $i$ so far, and let the global best $G$ be the best Q-values found by all the agents so far. Each agent updates its Q-values by using these two kinds of best Q-values. Following the update procedures of PSO, the update equations of a Q-value are given by

$$\begin{aligned} V_i(s,a) \leftarrow\ & WV_i(s,a) + C_1R_1\{P_i(s,a) - Q_i(s,a)\} \\ & +C_2R_2\{G(s,a) - Q_i(s,a)\} \end{aligned} \tag{5}$$

$$Q_i(s,a) \leftarrow Q_i(s,a) + V_i(s,a) \tag{6}$$

where $V_i(s,a)$ is a so-called velocity, $W$, $C_1$ and $C_2$ are weight parameters, and $R_1$ and $R_2$ are uniform random numbers in the range from 0 to 1.

## 3 Proposed Swarm Reinforcement Learning Algorithm

In this section, we point out a problem of SRL-PSO, and propose SRL-PSO-L, an extension of SRL-PSO, in order to resolve the problem.

In SRL-PSO, overvalued Q-values sometimes become the personal best. Q-values of the agent whose personal best is overvalued are not updated adequately, which causes serious deterioration of learning speed. As mentioned in Subsection 2.2, Q-values of each agent are evaluated based on rewards gained during the previous one episode. The rewards gained for the Q-values are variational because of stochastic transition of states. Therefore, even if the Q-values are considerably different from the Q-values which bring the maximum return, they are evaluated to be much superior in the case where the agent unexpectedly gains large rewards, which is an inevitable problem in SRL-PSO. If such overvalued Q-values become the personal best or the global best, then they are mostly superior to Q-values updated by the individual Q-learning and they are unfortunately kept to be the personal best or the global best. Since the Q-values of agent continue to be updated by using the overvalued Q-values in the learning through exchanging information, the learning speed deteriorates.

In SRL-PSO-L, if the personal best is not replaced after experiencing a specified number of episodes, it is mandatorily replaced in order not to store overvalued Q-values over many episodes. For this purpose, the concept of age and lifespan is introduced into the personal best. After an agent performs the individual Q-learning, the age of its own personal best increases by one. If the personal best is not replaced after experiencing a certain number of episodes corresponding to a specified lifespan, it comes the end of its life and its Q-values are replaced with the current Q-values of the agent. If it is replaced, its age is reset to zero. In addition, if the personal best which comes the end of its life is the global best, the global best is also replaced. In this case, the (new) personal bests of all the agents are compared, and the best among them is stored as the global best. By using this information exchange method, even if overvalued Q-values are the personal best, it is replaced after experiencing some episodes and it is expected that a better policy is found rapidly.

The flow of SRL-PSO-L is as follows.

**SRL-PSO-L**

$E_i$ : evaluated value for the Q-values $\{Q_i(s,a)\}$ of agent $i$.
$E_i^{\mathrm{P}}$ : evaluated value for the personal best $\{P_i(s,a)\}$ of agent $i$.
$E^{\mathrm{G}}$ : evaluated value for the global best $\{G(s,a)\}$.
$i^{\mathrm{G}}$ : agent number in which the Q-values are the origin of the global best.
$age_i$ : age of agent $i$.
$L$ : lifespan of the personal best.
$Y$ : number of episodes for which the individual Q-learning is performed between the information exchange among the agents.

```
Step 1. $\forall i, s, a$: set the initial values of $Q_i(s,a)$ and $V_i(s,a)$, and set $age_i \leftarrow 0$,
        $E_i^{\mathrm{P}} \leftarrow -\infty$ and $E^{\mathrm{G}} \leftarrow -\infty$.
Step 2. Perform the following procedures for each agent $i$.
  Step 2-1. Set $y \leftarrow 1$. The variable $y$ means the number of episodes after the
            information exchange.
  Step 2-2. Update $Q_i(s,a)$ by performing the individual Q-learning (1) for one
            episode.
  Step 2-3. Set $age_i \leftarrow age_i + 1$. If $age_i = L$, go to Step 2-7.
```

Step 2-4. Calculate the evaluated value $E_i$ by (2). If $E_i > E_i^{\mathrm{P}}$, $\forall s, a$: set $P_i(s,a) \leftarrow Q_i(s,a)$, $E_i^{\mathrm{P}} \leftarrow E_i$ and $age_i \leftarrow 0$.
Step 2-5. If $E_i > E^{\mathrm{G}}$, $\forall s, a$: set $G(s,a) \leftarrow Q_i(s,a)$, $E^{\mathrm{G}} \leftarrow E_i$ and $i^{\mathrm{G}} \leftarrow i$.
Step 2-6. Go to Step 2-9.
Step 2-7. $\forall s, a$: set $P_i(s,a) \leftarrow Q_i(s,a)$, $E_i^{\mathrm{P}} \leftarrow E_i$ and $age_i \leftarrow 0$.
Step 2-8. If $i = i^{\mathrm{G}}$, $\forall s, a$: set $i^{\mathrm{G}} \leftarrow \arg\max_i E_i^{\mathrm{P}}$, $G(s,a) \leftarrow P_{i^{\mathrm{G}}}(s,a)$ and $E^{\mathrm{G}} \leftarrow E_{i^{\mathrm{G}}}^{\mathrm{P}}$.
Step 2-9. If $y < Y$, set $y \leftarrow y + 1$ and return to Step 2-2.
Step 3. Update $V_i(s,a)$ and $Q_i(s,a)$ of each agent $i$ by (5) and (6).
Step 4. If the termination condition is satisfied, terminate this algorithm. Otherwise, return to Step 2.

## 4 Numerical Experiments

The effectiveness of the proposed SRL-PSO-L is examined by applying it to a shortest path problem and by comparing its computational efficiency with that of other algorithms. A usual shortest path problem may easily be solved by ordinary Q-learning, and we make it harder to solve. We set up a shortest path problem with the goal position being changing randomly.

### 4.1 Shortest Path Problem

The shortest path problem is that a single agent finds the shortest path from the start cell to the goal cell in an $n$ by $n$ grid world. In this grid world, we let the coordinates at the bottom left be (1,1), and those at the top right be ($n$,$n$). While the start cell is (1,1) and fixed, the goal cell is changing within a range and is determined at random. The agent perceives its own coordinates ($x$,$y$), and has four

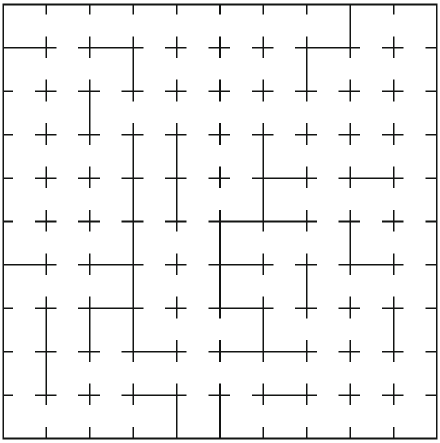


**Fig. 1.** Example of grid world for $n$=10

possible actions to take: moving up, moving down, moving left and moving right, that is to say, it has actions to move into $(x, y + 1)$, $(x, y - 1)$, $(x - 1, y)$ and $(x + 1, y)$. Some cells have walls at the boundaries with their adjacent cells, and the movement of the agent is blocked by the walls and the edge of the grid world.

Figure 1 shows an example of the grid world for $n$=10. In our experiments, we let the size $n$=40, and the $x$ and $y$ coordinates of the goal cell are determined randomly in the range from 35 to 40, respectively. Hence, they are determined randomly from 36 cells. Four cases 1, 2, 3 and 4 in which the number and positions of the walls are different among them are generated according to the above condition. They are numbered in the ascending order of the number of walls, and there is no wall in Case 1.

### 4.2 Experimental Set Up

In this paper we consider problems of single-agent tasks, and the shortest path problem explained in the previous subsection is a single-agent task. SRL-PSO-L, SRL-PSO and ordinary Q-learning with a single agent (called QL) are applied to this problem, and their experimental results are compared. The following values are used for parameters of Q-learning:

learning-rate parameter : $\alpha = 0.5$,
discount-rate parameter : $\gamma = 0.999$,
probability of random action : $\varepsilon = 0.2$.

All the initial values of $Q_i(s, a)$ are set to be zero. The coordinates of the goal cell are changed randomly at the beginning of every episode, and are fixed during the episode. When the agent reaches the goal cell, it gains the reward +100. Otherwise, it gains −1 whenever it takes an action. By using this rule of rewarding, the evaluated value $E$ becomes larger when the number of actions to reach the goal cell is smaller.

For SRL-PSO-L and SRL-PSO, the following values are also used for their parameters:

number of agents : 4,
weights in (5) and (6) : $W = 0, C_1 = C_2 = 0.2$,
number of episodes for which the individual Q-learning is performed between the information exchange : $Y = 1$,
discount parameter in (2) : $d = 0.999$,
lifespan : $L = 50$ (for SRL-PSO-L).

The other experimental conditions are the same as QL.

Each algorithm is terminated after reaching 20000 episodes. In SRL-PSO-L and SRL-PSO, since the number of agents is four, each agent learns for 5000 episodes. The values of all the parameters are determined through the preliminary experiments in such a way that each algorithm works as good as possible.

### 4.3 Results and Discussion

Figure 2 shows the variation of number of actions through the learning phase obtained by each algorithm for Case 4, which is the most complicated case because of many walls. In SRL-PSO-L and SRL-PSO, the x-axis is not the number of episodes for each single agent, and is the sum of numbers of episodes for all the agents. The y-axis represents the minimum number of actions to reach the goal found so far. Each algorithm is performed thirty times with various random seeds, and their results are averaged.

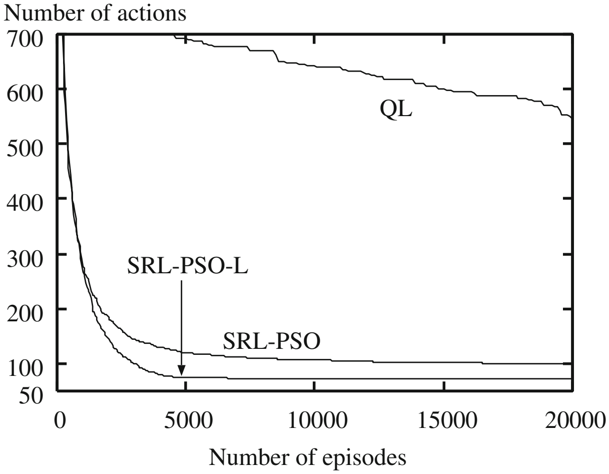


**Fig. 2.** Variation of number of actions through the learning phase (Case 4)

It is observed from the figure that the numbers of actions in SRL-PSO-L and SRL-PSO are smaller than that of QL. Therefore, learning with the multiple agents works better. Comparing SRL-PSO-L with SRL-PSO, SRL-PSO-L is better than SRL-PSO, and a better policy is obtained rapidly by introducing the lifespan into the personal best.

Next, simulations are performed by using Q-values which each algorithm found through the previous learning phase, and the algorithm is evaluated by the number of actions to reach the goal cell in the simulations. Since there exist 36 coordinates of the goal cell in this shortest path problem, the simulations are performed ten times for each of these goal coordinates and their results are averaged. The $\varepsilon$-greedy method is used for taking an action, and the probability

**Table 1.** Comparison of number of actions in the simulations performed by using the Q-values obtained

| Case | SRL-PSO-L | SRL-PSO | QL |
|---|---|---|---|
| 1 | 261 | 287 | 945 |
| 2 | 319 | 321 | 915 |
| 3 | 515 | 599 | 996 |
| 4 | 459 | 574 | 999 |

of random action is 0.2, which is the same value as the learning phase. Table 1 shows the mean number of actions to reach the goal cell in the simulations. It is confirmed from this table that SRL-PSO-L outperforms SRL-PSO and QL.

## 5 Conclusion and Future Works

This paper has pointed out a problem of SRL-PSO, and has proposed an extension of SRL-PSO in which a lifespan is introduced into the personal best. In the proposed method, if the personal best is not replaced after experiencing episodes corresponding to the lifespan, it comes the end of its life and its Q-values are replaced with the current Q-values. In addition, if the personal best which comes the end of its life is the global best, the global best is also replaced. It is confirmed from the experimental results that the proposed algorithm using the lifespan outperforms SRL-PSO and ordinary Q-learning using a single agent.

As we pointed out, in SRL-PSO, the evaluated values of Q-values inevitably have errors because they are approximates to the returns. The evaluated values in the swarm reinforcement learning method correspond to objective function values in optimization, and these errors are regarded as noises of the objective function values. Extensions of PSO are proposed for optimization problems whose objective functions are noisy [7,8]. Investigating whether the concept of these extensions works well for SRL-PSO is one of our future works. In contrast, the concept of the proposed method, that is, introducing a lifespan into PSO could be effective for the optimization problems with the noisy objective functions, and developing such PSO is the other future work.

## References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning. MIT Press, Cambridge (1998)
2. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
3. Iima, H., Kuroe, Y.: Reinforcement Learning through Interaction among Multiple Agents. In: SICE-ICASE International Joint Conference, pp. 2457–2462 (2006)
4. Iima, H., Kuroe, Y.: Swarm Reinforcement Learning Algorithms Based on Particle Swarm Optimization. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 1110–1115 (2008)
5. Watkins, C.J.C.H., Dayan, P.: Q-Learning. Machine Learning 8, 279–292 (1992)
6. Busoniu, L., Babuska, R., Schutter, B.D.: A Comprehensive Survey of Multiagent Reinforcement Learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C 38, 156–172 (2008)
7. Pugh, J., Martinoli, A., Zhang, Y.: Particle Swarm Optimization for Unsupervised Robotic Learning. In: IEEE Swarm Intelligence Symposium, pp. 92–99 (2005)
8. Pugh, J., Martinoli, A.: Multi-Robot Learning with Particle Swarm Optimization. In: International Conference on Autonomous Agents and Multiagent Systems, pp. 441–448 (2006)

# Effectiveness of Intrinsically Motivated Adaptive Agent for Sustainable Human-Agent Interaction

Takayuki Nozawa and Toshiyuki Kondo

Tokyo University of Agriculture and Technology,
2-24-16 Naka-cho, Koganei-shi, Tokyo 184-8588, Japan
{tknozawa,t_kondo}@cc.tuat.ac.jp

**Abstract.** To achieve sustainable human-agent interaction (HAI), we proposed a new model of *intrinsically motivated adaptive agent*, which learns about the human partner and behaves to satisfy its intrinsic motivation. To investigate the model's effectiveness, we conducted a comparative HAI experiment with a simple interaction setting. The results showed that the model was effective in inducing subjective impressions of higher enjoyability and sustainability. The subjects' brain activity measured by near-infrared spectroscopy (NIRS) indicated higher variability of activity at left dorsolateral prefrontal cortex during the interaction with the proposed agent.

**Keywords:** Human-agent interaction (HAI); intrinsic motivation; adaptive agent; reinforcement learning; near-infrared spectroscopy (NIRS).

## 1 Introduction

Research on human-agent interaction (HAI) and human-robot interaction (HRI) has recently been growing and producing a wide range of applications, such as entertainment, therapeutic use, media of communication, and other kinds of assistance [1]. For many of these applications, continuation of interaction and maintaining the user's interest are the issues of primary importance.

Among the various factors which affect the sustainability of interaction, Nakata *et al.* focused on the information transmission between agent and human. They experimentally compared agents with different degrees of regularity, and showed that maximum human interest is achieved by interaction with the agent of intermediate regularity [2,3]. Similarly, Kondo *et al.* investigated the relationship between predictability and sustainability of the interaction, and showed that moderate level of predictability can contribute to the sustainability [4].

However, humans get bored even with agents of moderate predictability, once they fix their mental model about the agents. To achieve more sustainable HAI, it will be useful to take notice of our own motivation in HAI, as well as in interaction with other human or animal: We are generally driven to continue interaction when it satisfies our *intrinsic motivation* for exploration, manipulation, achievement, etc [5]. Therefore, to make HAI more "natural" and sustainable, a promising approach would be to endow the agent with the intrinsic motivation.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 179–188, 2009.

Intrinsic motivation has recently been utilized in developmental robotics (also known as epigenetic robotics or ontogenetic robotics) to learn progressively from simpler to more complex situations, avoiding situations in which nothing can be learned [6,7]. However, effectiveness of intrinsically motivated agent on HAI is not obvious, because unlike the usual cases of developmental robotics, the environment (human partner) can be highly dynamic. In this study, we propose a new model of adaptive interaction agent which learns about the human partner and behaves to satisfy its intrinsic motivation. We implemented the agent for a simple interaction setting, and conducted a comparative experiment to investigate the model's effectiveness for enjoyable and sustainable HAI. In addition to investigating subjective impressions of the agents and interaction log, we measured the subjects' brain activity during interaction by near-infrared spectroscopy (NIRS), to study the effect of different types of agents on the subjects' cognitive states.

## 2 Model of Intrinsically Motivated Agent

### 2.1 Adaptivity and Reinforcement Learning

We focus on turn-taking type of interaction, which means that an interaction is described from the agent's viewpoint as a sequence

$$\cdots \to s_t \to a_t \to s_{t+1} \to a_{t+1} \to \cdots \ , \tag{1}$$

where $s_t \in \mathcal{S}$ denotes *sensory input* from the human partner to the agent at time $t$, and $a_t \in \mathcal{A}$ denotes *action* of the agent at $t$. We presume that the set $\mathcal{A}$ of possible agent actions at $t$ is not restricted by the preceeding input $s_t$.

An agent driven by certain motivation system — whether intrinsic or extrinsic — must be able to adapt to the environment (partner), to satisfy its motivation. We use the TD learning method [8] to model an adaptive agent.

In the framework of reinforcement learning, each input $s_t$ is accompanied by a *reward* $r_t \in \mathbb{R}$. When a new input $s_{t+1}$ is obtained, the agent updates *value* of the preceeding input $s_t$ based on the stored reward $r_t$ and the current *value function* $V : \mathcal{S} \to \mathbb{R}$, as

$$V(s_t) \leftarrow V(s_t) + \alpha\{r_t + \gamma V(s_{t+1}) - V(s_t)\} \ . \tag{2}$$

Here $\alpha$ is learning rate parameter of the value function, and $\gamma$ is discount rate of a future reward in the present value. In the initial state, without any a priori knowledge, one can set $V(s) = 0$ for all $s \in \mathcal{S}$.

Based on the value function $V$, the agent selects an action which is likely to maximize expected values for the next moment. Detail of the action selection is given in Sect. 2.2.

### 2.2 Internal Model of Human Partner

In many simple learning problems, the reward $r_t$ is directly associated with $s_t$. In the intrinsically motivated agent, however, $r_t$ is derived from the agent's *internal model* of the environment (partner). The role of the internal model is to predict

– what will be the next input $s_{t+1}$ given a *context* $(s_t, a_t)$,[1] and
– how it is likely that a contextual situation $(s_t, a_t)$ itself will take place.

For some specific problems, one could construct parametric internal models. Here, we adopt more extensive approach and define the internal model as a combination of the transition probability distribution $\{P_T(s_{t+1}|s_t, a_t)\}$ and the context probability distribution $\{P_C(s_t, a_t)\}$.

The internal model is updated based on the interaction history (1). As mentioned earlier, human partner can be highly dynamic. Therefore, the update of the internal model should incorporate decay of the memory. When the current context $(s_t, a_t)$ and the new input $s_{t+1}$ are given, the transition probability distribution $\{P_T\}$ is updated by

$$P_T(s'|s_t, a_t) \leftarrow P_T(s'|s_t, a_t) + \rho_T\{\delta(s', s_{t+1}) - P_T(s'|s_t, a_t)\} , \qquad (3)$$

where $\delta(x, y)$ is Kronecker's delta. Similarly, the context probability distribution $\{P_C\}$ is updated by

$$P_C(s, a) \leftarrow P_C(s, a) + \rho_C\{\delta((s, a), (s_t, a_t)) - P_C(s, a)\} . \qquad (4)$$

$\rho_T$ and $\rho_C$ are learning rates of the internal model. In the initial state, one can set $P_T(s'|s, a) = 1/|\mathcal{S}|$ for all $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ and $P_C(s, a) = 1/|\mathcal{S} \times \mathcal{A}|$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$.

The transition probability $\{P_T\}$ works also in the action selection to derive *action values* in TD learning as

$$V(a_{t+1} = a|s_{t+1}) = \sum_{s'} V(s')P_T(s_{t+2} = s'|s_{t+1}, a_{t+1} = a) . \qquad (5)$$

We utilize the Boltzmann action selection method, so the probability of the agent selecting action $a_{t+1} = a$, given $s_{t+1}$, is

$$p(a_{t+1} = a|s_{t+1}) = \frac{e^{V(a|s_{t+1})/T}}{\sum_b e^{V(b|s_{t+1})/T}} , \qquad (6)$$

where $T$ is the temperature parameter that determines the balance between the maximization of the expected value based on the current value function and the exploration for refinement of the value function.

### 2.3 Intrinsic Motivation for Information Transfer

In considering the proper expression of reward for the intrinsically motivated agent, we direct our attention on the *transfer entropy* [9], which is a information theoretic measure quantifying the causal interaction between two systems, excluding the shared information due to common history. The transfer entropy can be utilized to characterize autonomous systems [10].

[1] Extension to context with longer history length is straightforward. However, longer history requires exponentially longer interaction to obtain stable internal model.

From the probability distribution $p$ of triadic interaction sequence $(s_t, a_t, s_{t+1})$, transfer entropy from the agent (A) to the human partner (H) is given as

$$\begin{aligned} TE_{A\to H} &= MI(s_{t+1}; a_t|s_t) = H(s_{t+1}|s_t) - H(s_{t+1}|s_t, a_t) \\ &= \sum_{s_t}\sum_{a_t}\sum_{s_{t+1}} p(s_{t+1}|s_t, a_t)p(s_t, a_t) \log \frac{p(s_{t+1}|s_t, a_t)}{p(s_{t+1}|s_t)} \ . \end{aligned} \tag{7}$$

The more the agent's action $a_t$ has influence on the human's response $s_{t+1}$ given the same $s_t$, the larger $TE_{A\to H}$ becomes (in other words, the more information the agent can transfer to the human partner).

Our intrinsically motivated agent tries to maximize this measure of influence $TE_{A\to H}$ on the human partner. This leads the following reward function

$$r_{t+1} = \log \frac{P_T(s_{t+1}|s_t, a_t)}{P_{C,T}(s_{t+1}|s_t)} = \frac{P_T(s_{t+1}|s_t, a_t)}{\sum_b P_T(s_{t+1}|s_t, b)P_C(s_t, b)/\sum_b P_C(s_t, b)} \ . \tag{8}$$

Note the reward can be calculated from the internal model $(\{P_T\}, \{P_C\})$. By replacing all the probability terms in Eq. (7) with those of the internal model, one can also obtain the agent's *subjective transfer entropy.*

Let us describe through several possible situations how the reward (8) controls the behavior of the agent. First, when the agent finds an action $a_t$ which can effectively induce otherwise rare response $s_{t+1}$ given $s_t$, $P_T(s_{t+1}|s_t, a_t) \gg P_{C,T}(s_{t+1}|s_t)$ and the reward is high. However, when the agent repeats the same pattern $(s_t, a_t, s_{t+1})$ for its high value, both the numerator and denominator terms of (8) come close to 1 due to the update rules (3) and (4), so the reward and the value decrease, making the agent stop the repetition. On the other hand, when $a_t$ is followed by "unexpected" $s_{t+1}$ given $s_t$ (that is, $P_T(s_{t+1}|s_t, a_t) < P_{C,T}(s_{t+1}|s_t)$), the reward becomes negative, and such things happen more often in the situations where the agent cannot influence the human response. Taking these cases together, one can expected that the reward makes the agent pursue intermediate level of novelty, avoiding situations in which nothing can be learned, in a similar way as the *intelligent adaptive curiosity* proposed by Oudeyer *et al.* [7].

### 2.4 Algorithm

Combining the components described above, the algorithm of the intrinsically motivated agent can be described in the following procedural form:

- Initialize $(s_0, a_0, \delta_0)$, the value function $V$, and the internal model $(\{P_T\}, \{P_C\})$;
- Starting from $t = 0$, repeat
  - With the given context $(s_t, a(t))$, obtain new input $s_{t+1}$ from the environment (partner);
  - Update the value function $V(s_t)$ by the TD learning method (2);
  - Update the internal model $(\{P_T\}, \{P_C\})$ by the rules (3) and (4);
  - Evaluate the reward $r_{t+1}$ by (8), and store it for the value function update (2) in the next time step;
  - Selects the action $a_{t+1}$ using the rules (5) and (6);
  - $t \leftarrow t + 1$;

## 3 Experiment

To assess the effectiveness of our model of intrinsically motivated adaptive agent on HAI, we conducted a comparative experiment of three types of agents in a simple interaction design.

### 3.1 Interaction Design

We used a virtual agent, rather than a real robot agent, to prevent physically induced artifacts on the NIRS measurement by minimizing the subjects' movements and changes in posture during interaction with the agent [11].

A CG image of the Sony AIBO was presented on a 14.1 inch LCD display that was placed 70 cm in front of the subject sitting on a chair. Using a computer mouse, the subject clicked or dragged on the agent image. Based on the mouse-button pressing time, the agent distinguished each mouse input either as a click or as a drag (thus $\mathcal{S} = \{\text{click}, \text{drag}\}$), with the threshold of 350 ms.

The agent, in return, showed one of four actions (movies) $\mathcal{A} = \{M_1, M_2, M_3, M_4\}$; Action $M_1$ was to move the agent's head upward, then downward, and upward back to neutral position. $M_2$ was to move its head upward and then shaking it left and right (once each side), then back to neutral position. $M_3$ was to move its head up-right, back to neutral, up-left, back to neutral again. $M_4$ was to move its head downward, wiggle it forward and backward three times in quick succession, then move back to neutral.[2] The meanings of the actions were left to the interpretation of each subject. Each action took 1.5 s, and the agent did not accept new mouse input till the period ends.

### 3.2 Agents in Comparison

The following three types of agents were compared in the interaction experiment:

**Type I** was the intrinsically motivated adaptive agent defined in Sect. 2.

**Type F** was an adaptive agent which has the same rules for the update of value function (2) and the internal model (3), (4), and for action selection (5), (6), but the reward was extrinsically given by either of the fixed functions

$$r_{t+1}^{F1} = \begin{cases} 1 & \text{if } s_{t+1} = \text{click}, \\ 0 & \text{if } s_{t+1} = \text{drag}, \end{cases} \quad \text{or} \quad r_{t+1}^{F2} = \begin{cases} 0 & \text{if } s_{t+1} = \text{click}, \\ 1 & \text{if } s_{t+1} = \text{drag}. \end{cases} \qquad (9)$$

**Type R** was a agent which selects its action $a_t$ randomly with equal probability $1/4$, regardless of the input $s_t$.

Following parameter values were used for the adaptive agents of type I and F; the learning rate of the value function $\alpha = 0.1$, the discount rate of the future reward $\gamma = 0.3$, the learning rate of the internal model $\rho_T = \rho_C = 0.1$, and the temperature parameter for the Boltzmann action selection $T = 1/30$. A preliminary experiment as well as some inter-agent simulation confirmed that the model's qualitative behavior is not altered by minor changes in the parameters.

[2] The action movies can be downloaded from our website, `http://www.livingsys.lab.tuat.ac.jp/~nozawa/temp/IMAgentMovies.zip`

### 3.3 Subjects and Procedures

Twelve healthy volunteers (all male, all right handed, mean age 22.6 ± 2.0 years) participated in the experiment. All subjects were explained about the experiment before giving written informed consent. This study was approved by the ethics committee of the Tokyo University of Agriculture and Technology.

Before the experiment, the subjects were familiarized with the operation, by 5 min practice session with an agent which showed all the four actions in an order when the mouse was clicked, and in the reverse order when dragged.

The subjects were instructed to freely set and change their aims of interaction. Each subject had interaction sessions with all the three types of the agents. The subjects were divided into four groups (three subjects each), by the two orderings of agent types, (R, I, F) or (F, I, R), and by the two kinds of reward $r^{F1}$ or $r^{F2}$ in (9) for type F agent.

Each interaction session consisted of 1 min fixation phase, 15 min interaction phase, and again 1 min fixation phase. During the fixation phases, the subjects were instructed to fixate their attention on the cross mark which was shown at the center of the LCD display. Each interaction session was followed by 10 min rest period, during which subjects were asked to complete the questionnaire.

### 3.4 Observation

**Questionnaire.** After each session, the subjects were asked to describe their impression about the agent they interacted with. After the second and third sessions, the subjects were also asked to answer a Likert scale questionnaire comparing the last two agents they intereacted with. The questionnaire consisted of 16 items with eight viewpoints, each item with seven rating levels from "strongly disagree" (−3) to "strongly agree" (+3). The eight viewpoints were: (1) enjoyable, (2) charming, (3) lively, (4) soothing, (5) consistent, (6) obedient, (7) insightful to your intention, and (8) desirable for long interaction. For each of the eight viewpoints, two items — "The latter felt *more* {adjective} than ther former." and "The latter felt *less* {adjective} than ther former." — were presented. This was to balance the influence of positive/negative expressions, and to check the consistency of each subject's ratings. The items were arranged in a randomized order.

**Interaction Log.** In regard to the subjects, timing and types of mouse operations were recorded. For the agents, timing and types of movie actions were recorded. For the adaptive agent of type I and F, the sequence of rewards $r_t$, the estimated values $V$, and the internal model $(\{P_T\}, \{P_C\})$ were also logged.

From these data, we examined statistics of human/agent actions, information theoretic measures of interactions between the subjects and agents, such as mutual information, distinguishability, controllability (dyadic) [3] and transfer entropy (triadic), and dynamics of these measures.

**NIRS Measurement.** Prefrontal region of human brain has been suggested to play significant roles for attention control, working memory, executive function, etc. [12], which will be important for sustainable HAI. Therefore, we measured the subjects' brain activity in prefrontal region during the interaction sessions.

Changes in the concentration of oxy- and deoxy-Hb were measured by a multichannel NIRS imaging system (FOIRE-3000, Shimadzu Co., Japan). We placed 8 source and 7 detector optodes, covering prefrontal regions (including Fp1, Fp2 and Fz positions of the international 10-20 system), by 22 channels.

To avoid physically induced artifacts, the subjects were asked to assume a preferred posture and retain it for the entire duration of the experiment. The subjects kept the NIRS optodes on throughout all the three interaction sessions.

# 4 Results

Due to the limitations of space, we show only a part of the results. One subject was removed from the analysis as he did not use mouse click in a whole session and rarely used click in the other two sessions.

## 4.1 Subjective Impressions

Figure 1 shows the distribution of the comparative ratings between type I and the other two types, with respect to the eight viewpoints given in above. The viewpoints with which Wilcoxon signed rank test (null hypothesis: the rating is symmetric about 0) showed significant difference with level $p < 0.05$ are marked with "*", and those with level $p < 0.01$ are marked with "**". This result shows that the intrinsically motivated adaptive agent gave impressions of higher enjoyability (viewpoint 1) and sustainability (viewpoint 8) than the other two agents.

## 4.2 Characteristics of Information Transfer

To compare the features of interaction with the different types of agents from a static viewpoint, we derived probability distributions of dyadic pairs $(s_t, a_t)$, $(a_t, s_{t+1})$, and triadic interactions $(a_{t-1}, s_t, a_t)$, $(s_t, a_t, s_{t+1})$, by counting over

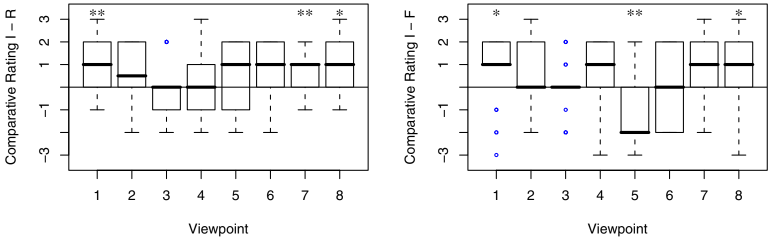


**Fig. 1.** Boxplot of the ratings on comparative impressions with respect to eight viewpoints. *Left*: the agent type I vs R. *Right*: the agent type I vs F. See Sect. 3.4 for the contents of the viewpoints.

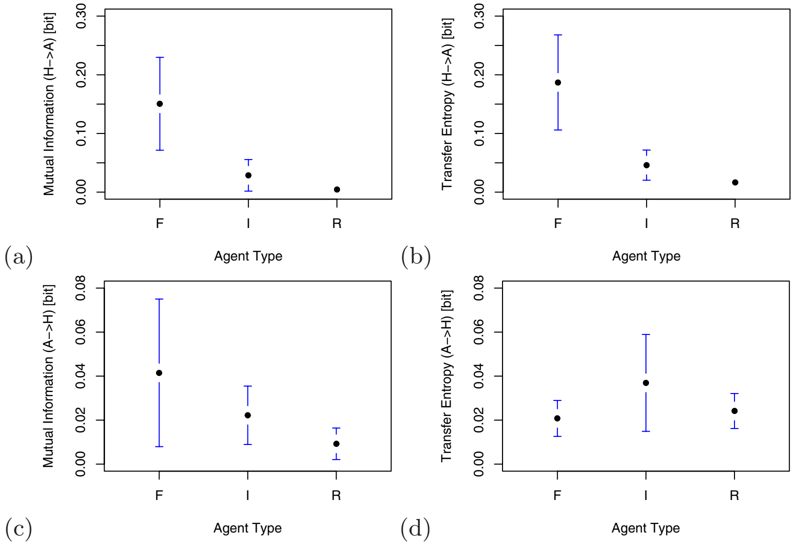


**Fig. 2.** Static estimation of (a) mutual information $MI(s_t, a_t)$, (b) transfer entropy $TE_{H\to A}$, (c) mutual information $MI(a_t, s_{t+1})$, (d) transfer entropy $TE_{A\to H}$

the whole of each interaction session. Using these, we calculated mutual information and transfer entropy and compared them among the agent types (Fig. 2).

Fig. 2(a) and (b) show that type I agent caused intermediate level of information transfer from the subject to the agent, meaning that the predictability for human was also intermediate between the other two types of agents. Comparison of Fig. 2(c) and (d) suggests that once $s_t$ was given, the action $a_t$ of type I agent had more influence on the human response $s_{t+1}$ than that of type F agent. We also note that the transfer entropy $TE_{A\to H}$ was significantly larger than the mutual information $MI(a_t, s_{t+1})$ (two-tailed paired $t$-test, $p < 0.05$). These results are consistent with the definition that the intrinsically motivated agent tries to maximize the influence $TE_{A\to H}$ on the human partner.

### 4.3 Variability of Prefrontal Cortex

We evaluated the variability of activity in the prefrontal region by the standard deviation of band-pass filtered[3] oxy-Hb signals from each of the 22 channels.

By multiple pairwise comparisons (Wilcoxon signed rank test with Holm's multiple test correction, $p < 0.05$), the variability showed significantly higher values with type I agent at a lower left channel (the data is shown in Table 1). The channel with the significantly higher variability was overlapped with the dorsolateral prefrontal cortex (DLPFC), which is involved in sustaining attention [12]. Therefore, this higher variability suggests that the intrinsically motivated adaptive agent successfully kept influencing on the subjects' attention level.

[3] 4-th order Chebyshev type II filter with cut-off frequencies 0.7 and 0.002 Hz, passband ripple 5 dB.

**Table 1.** Variability of NIRS oxy-Hb signal (a.u.) from a lower left channel

| | Subject | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| I | 2.00 | 5.53 | 1.93 | 3.94 | 2.83 | 4.40 | 3.45 | 6.54 | 3.20 | 1.45 | 0.89 |
| Type F | 1.65 | 4.78 | 1.41 | 1.80 | 2.47 | 3.41 | 2.29 | 3.40 | 2.91 | 2.13 | 0.93 |
| R | 1.21 | 5.13 | 1.83 | 3.09 | 2.60 | 3.98 | 2.45 | 5.63 | 1.50 | 1.60 | 0.76 |

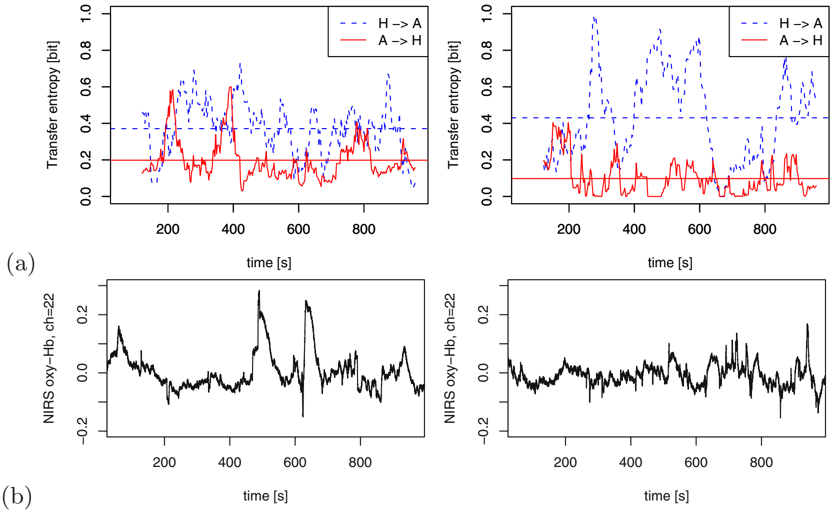


**Fig. 3.** Dynamical characteristics of the interactions by a subject with agents of type I (left) and F (right). (a) dynamics of transfer entropy from human to agent (dashed) and agent to human (solid). Horizontal lines shwo average values of the corresponding quantities. (b) NIRS oxy-Hb signals at the lower left channel overlapped with DLPFC.

### 4.4 Dynamics of Interaction Characteristics

Dynamics of the informational characteristics brings further insight into the difference of the interactions. For example, Fig. 3(a) shows dynamics of transfer entropy $TE_{H \to A}$ and $TE_{A \to H}$ of interactions between a subject and type I/F agents. The values were calculated from triadic probability distributions obtained by counting over the interaction history in shifting time windows of 60 s. It indicates that type I agent had more initiative in the interaction than type F agent did. Figure 3(b) shows the same subject's NIRS oxy-Hb signals at the lower left channel overlapped with DLPFC. The results suggest that the activity of DLPFC was raised when the interaction became transitional and less stable.

## 5 Conclusions

To achieve sustainable HAI, we proposed a new model of intrinsically motivated adaptive agent, which tries to maximize its influence on the human partner. The comparative experiment showed that the model worked as expected, and was effective in inducing subjective impressions of higher enjoyability and sustainability. The subjects' brain activity measured by NIRS indicated higher variability of activity at left DLPFC during the interaction with the proposed agent.

Unlike the models of intrinsically motivated learner in developmental robotics [6,7], our model did not incorporated the extension of dimensions in input-action space and the internal model space. Such "developmental" aspect will be effective for longer term sutainable HAI, though it would make experimental assessment of effectiveness more difficult.

**Acknowledgments.** This study was partially supported by "Symbiotic Information Technology Research Project" of Tokyo University of Agriculture and Technology, and also by the Grant-in-Aid for "Scientific Research on Priority Areas (Area No. 454)" from the Japanese Ministry of Education, Culture, Sports, Science and Technology.

## References

1. Fong, T., Nourbakhsh, I., Dautenhahn, K.: A Survey of Socially Interactive Robots. Robotics and Autonomous Systems 42, 143–166 (2003)
2. Nakata, T., Ko, T.S., Mori, T., Sato, T.: Informational Analysis on Impression of Human Robot Interaction. J. Robotics Soc. Japan 19, 667–675 (2001) (in Japanese)
3. Nakata, T.: Information Transmission and Impression Formulation in Human-Robot Interaction, http://staff.aist.go.jp/toru-nakata/hri/HRI-Info.html
4. Kondo, T., Hirakawa, D., Nozawa, T.: Sustainability and Predictability in a Lasting Human-Agent Interaction. In: Prendinger, H., Lester, J.C., Ishizuka, M. (eds.) IVA 2008. LNCS (LNAI), vol. 5208, pp. 505–506. Springer, Heidelberg (2008)
5. Deci, E.L., Ryan, R.M.: Intrinsic motivation and self-determination in human behavior. Plenum, New York (1985)
6. Barto, A., Singh, S., Chentanez, N.: Intrinsically motivated learning of hierarchical collections of skills. In: 3rd International Conference on Developmental Learning, pp. 112–119. UCSD Institute for Neural Computation, La Jolla (2004)
7. Oudeyer, P., Kaplan, F., Hafner, V.V.: Intrinsic Motivation Systems for Autonomous Mental Development. IEEE Trans. Evol. Comput. 11, 265–286 (2007)
8. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
9. Schreiber, T.: Measuring Information Transfer. Phys. Rev. Lett. 85, 461–464 (2000)
10. Bertschinger, N., Olbrich, E., Ay, N., Jost, J.: Autonomy: An information theoretic perspective. BioSystems 91, 331–345 (2008)
11. Nozawa, T., Kondo, T.: A comparison of artifact reduction methods for real-time analysis of fNIRS data. In: Salvendy, G., Smith, M.J. (eds.) HCII 2009. LNCS, vol. 5618, pp. 413–422. Springer, Heidelberg (2009)
12. Miller, E.K., Cohen, J.D.: An Integrative Theory of Prefrontal Cortex Function. Annu. Rev. Neurosci. 24, 167–202 (2001)

# RAST: A Related Abstract Search Tool

Shiro Usui[1,2], Nilton L. Kamiji[1], Tatsuki Taniguchi[3], and Naonori Ueda[4]

RIKEN Brain Science Institute
[1] Laboratory for Neuroinformatics
[2] Neuroinformatics Japan Center, 2-1 Hirosawa, Wako, Saitama, 351-0198, Japan
[3] IVIS Inc., Tokyo, Japan
[4] NTT CS-Labs., Kyoto, Japan
usuishiro@riken.jp, nilton@brain.riken.jp,
tatsuki.taniguchi@ivis.co.jp, ueda@cslab.kecl.ntt.co.jp

**Abstract.** With the increasing amount of information available in recent years, searching for the desired content is becoming a challenging task. In this work, a tool for searching abstracts submitted to scientific conferences is introduced. It not only search abstracts by the given keyword(s), but also displays abstracts related to a single or multiple selection. It also displays information on the similarity between all abstracts on a list, providing users with more information to take into account to support finding relevant abstracts. It also suggests possible keywords to help refine their search, and an interface for storing and sorting selected abstracts for future review and/or printing.

**Keywords:** text data mining, knowledge visualization, neuroinformatics, bipartite graph, dimensionality reduction.

## 1 Introduction

As more and more information is archived every day, techniques to efficiently extract essential information are required. When dealing with textual documents, a widely used approach is the vector space model [1]. In this model, a set of terms T is build by extracting words from the collection of documents D followed by stop words removal and stemming [2]. In this approach, the numbers of occurrences of each term in each document are counted (document frequency), and a matrix F is built with one row for each document and one column for each term and the document frequency as entries.

In order to grasp the overall picture of the dataset being analyzed, several visualization techniques have been proposed [3--5]. Based on these concepts, we have developed a text mining and visualization tool utilizing the spherical embedding (SE) algorithm [6--7]. Moreover, as an application of such a tool, we have also developed a tool for searching and visualizing documents and contents of databases [8--9].

In this work, a tool for searching related abstract submitted to scientific conferences is introduced. It does not provide a visualization of the overall dataset previously introduced [8], however, it has an interactive tool for learning about the similarity between all abstracts on a list. Such a feature may help users in finding relevant abstracts. It also displays possible keywords candidates so the user can use to

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 189–195, 2009.

refine their search. Moreover, a feature for managing a collection of selected abstracts is also made available for future reviewing and/or printing.

## 2 Concepts of the Related Abstract Search Tool

The Related Abstract Search Tool (RAST) is a web-based application for searching scientific abstracts submitted to conferences such as the Society for Neuroscience Annual Meeting, Human Brain Mapping, Japanese Neuroscience Society meeting, and so on. It was first implemented as an extension of the 3D-SE viewer [6--9], which is a tool for visualizing the relationship between two entities: words and abstracts.

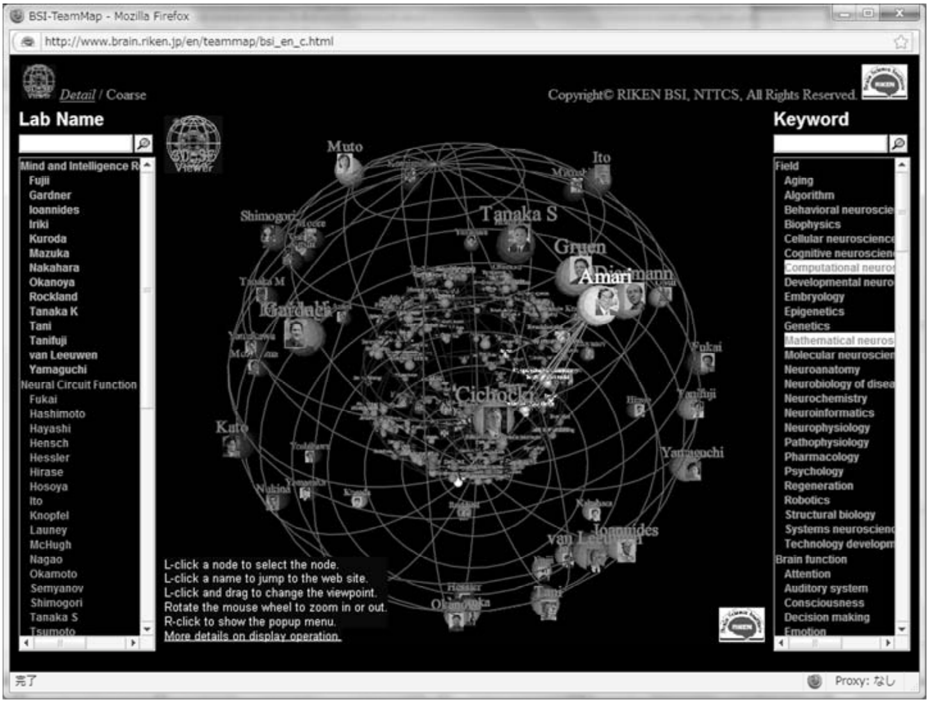
### 2.1 The 3D-SE Viewer

The 3D-SE viewer is a tool for the visualization of bipartite graphs [6--7]. As one of its application, we have developed the RIKEN Brain Science Institute team map (BSI-Team Map) [10], which shows the relationship between laboratories and units in BSI and the keywords representing their research topics (Fig. 1). The information provided for the creation of this output was only on the existence or absence of a link between the two vertices. However, surprisingly, the 3D-SE viewer clearly clustered laboratories belonging to the same category (Research Cores). This result suggests that this tool can be utilized to cluster, visualize and browse over a dataset consisting of two entities.

### 2.2 The 3D-SE Viewer Based RAST

As described above, the 3D-SE viewer may provide the overall image of a large dataset by visualizing the relationship between two entities [7]. We have therefore, applied it for the visualization of abstracts submitted to scientific conferences [8]. In this example [11], words were extracted [12] from abstracts of the Society for Neuroscience 2008 annual meeting poster sessions, and the coordinates of the abstracts and extracted keywords calculated using the 3D-SE viewer. This dataset consists of 14,029 abstracts and 47,868 extracted words. To visualize it using the 3D-SE viewer would require powerful computational resources, therefore, viewing the sphere as the earth, we have adopted the Google Maps API, compressing the three dimensional (3D) sphere into a two dimensional (2D) plane.

In this tool (Fig. 2), results of a keyword search are displayed not only as a list of abstracts (key abstracts), but also graphically by placing flags over the abstracts (nodes) in the map. The key feature of this tool is that, based on the similarity index calculated online, it provides a list of abstracts related to a single or multiple chosen key abstract(s) (related abstracts). More surprisingly, this feature could provide several users with interesting related abstracts which did not contain any keyword utilized for the key abstract search. We call them “hidden treasure”. Moreover, a list of extracted keywords is also provided for each listed abstract. Such information may provide users with candidates of keywords regarding their research interest. These keywords may be useful for searching information over the web, and also in other databases.



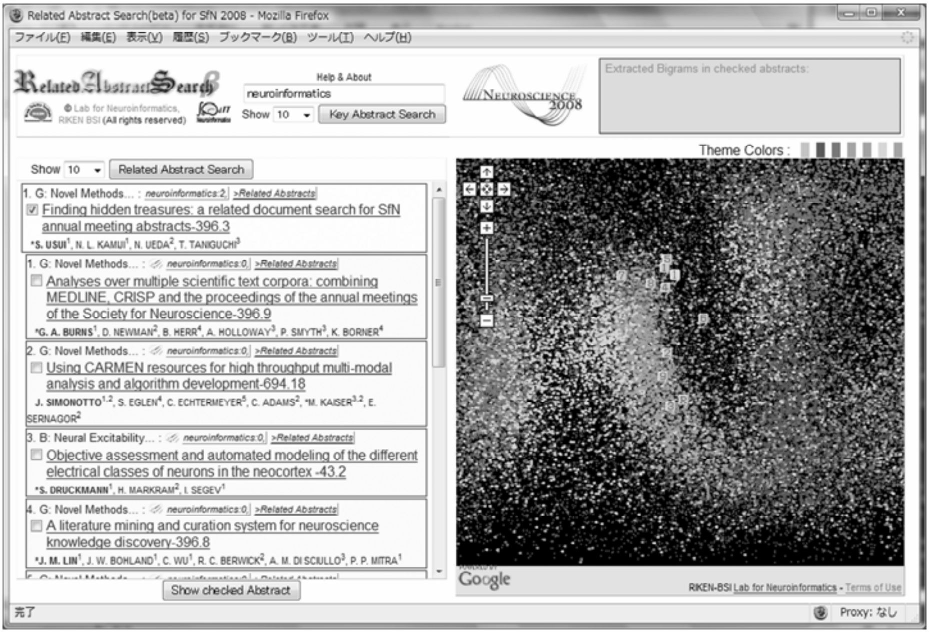
L-click a node to select the node.
L-click a name to jump to the web site.
L-click and drag to change the viewpoint.
Rotate the mouse wheel to zoom in or out.
R-click to show the popup menu.
More details on display operation.

**Fig. 1.** The BSI Team Map. The list on the left represents different laboratories colored on the group (Research Core) it belongs to. The list on the right represents the keywords describing a research topic colored into research categories such as "Field", "Brain Function", "Method", etc.

Similar feature is observed elsewhere as in the PubMed database, however, they do not allow search for multiple selected items. And since they can only search for related items to a single selection, the results might have been calculated offline and stored on a database to be referenced when a query arrives.

## 2.3 Problems on the 3D-SE Viewer Based RAST

As it can be noted from the examples cited above, the 3D-SE viewer is not only a tool for visualizing bipartite graph, but it also compresses multidimensional data into a three dimensional space. For example, in the BSI-Team map, there are 58 teams and 175 keywords, therefore, the 3D-SE viewer is compressing a 175 dimensional vector into three dimensions. Similarly, for the SfN 2008 RAST, it is compressing a much larger vector of 47,868 dimensions into only three. As a result of this process, not all abstracts closely located are directly related to each other. This has resulted in misunderstanding on the interpretation of the map. However, since again we have only provided information on the existence or absence of a link between abstracts and words, the grouping of abstracts belonging to the same category suggests that it does provide visual information of the overall characteristic of the data.


Related Abstract Search(beta) for SfN 2008 - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)
Related Abstract Search β
© Lab for Neuroinformatics, RIKEN BSI (All rights reserved)
Help & About
neuroinformatics
Show 10
Key Abstract Search
NEUROSCIENCE 2008
Extracted Bigrams in checked abstracts:
Theme Colors :
Show 10
Related Abstract Search
Show checked Abstract
Google
RIKEN-BSI Lab for Neuroinformatics - Terms of Use
完了
Proxy: なし

1. G: Novel Methods... : neuroinformatics:2 >Related Abstracts
☑ Finding hidden treasures: a related document search for SfN annual meeting abstracts-396.3
*S. USUI[1], N. L. KAMUI[1], N. UEDA[2], T. TANIGUCHI[3]

1. G: Novel Methods... : neuroinformatics:0 >Related Abstracts
□ Analyses over multiple scientific text corpora: combining MEDLINE, CRISP and the proceedings of the annual meetings of the Society for Neuroscience-396.9
*G. A. BURNS[1], D. NEWMAN[2], B. HERR[4], A. HOLLOWAY[3], P. SMYTH[3], K. BORNER[4]

2. G: Novel Methods... : neuroinformatics:0 >Related Abstracts
□ Using CARMEN resources for high throughput multi-modal analysis and algorithm development-694.18
J. SIMONOTTO[1,2], S. EGLEN[4], C. ECHTERMEYER[5], C. ADAMS[2], *M. KAISER[3,2], E. SERNAGOR[2]

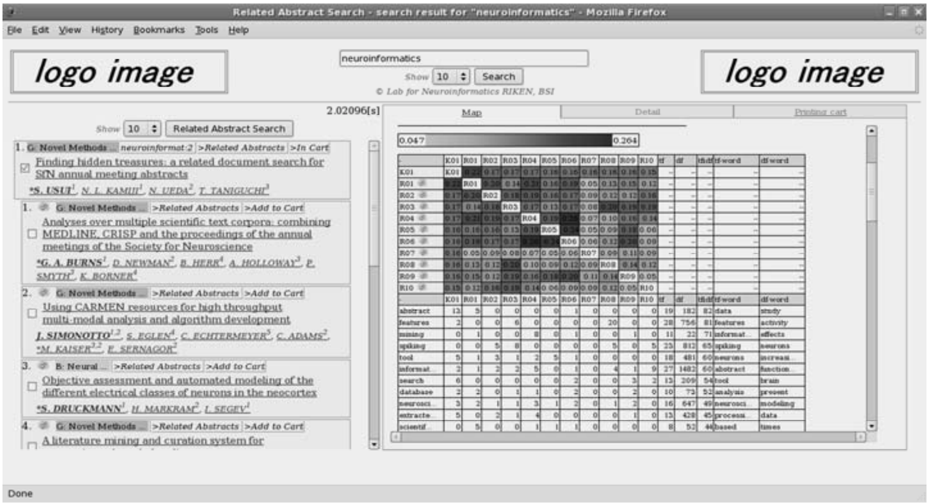
3. B: Neural Excitability... : neuroinformatics:0 >Related Abstracts
□ Objective assessment and automated modeling of the different electrical classes of neurons in the neocortex -43.2
*S. DRUCKMANN[1], H. MARKRAM[2], I. SEGEV[1]

4. G: Novel Methods... : neuroinformatics:0 >Related Abstracts
□ A literature mining and curation system for neuroscience knowledge discovery-396.8
*J. M. LIN[1], J. W. BOHLAND[1], C. WU[1], R. C. BERWICK[2], A. M. DI SCIULLO[3], P. P. MITRA[1]

**Fig. 2.** 3D-SE viewer based Related Abstract Search Tool. A list of abstracts is displayed on the left side, and the 3D-SE viewer sphere compressed into a 2D plane shown in the right. Yellow background on the list mean key abstract (yellow flags on the map), and blue background (blue flags) are the related abstracts. Most of the related abstracts did not utilize the term "neuroinformatics" entered for searching the key abstract.

## 3 The Related Abstract Search Tool

The RAST has been designed to search for related abstracts online based on single or multiple selected items. The base items, called key abstracts are first searched based on the keyword entered by the user. Key abstracts are listed in yellow background according to the number of found keywords (Fig. 3 left side). The number of items to be listed is selected from the pull down menu found before the "Search" button. Related abstracts are searched either by pressing the "> Related Abstract" link located at each abstract on the list, or by selecting the key abstracts by checking the check box in front of each abstract and pressing the "Search Related Abstract" button. Related abstracts are listed indented and below the key abstracts, in blue background. Similarly to the list of key abstracts, the number of items to be listed for the related abstracts can be selected from the pull down menu found before the button. On the right side of the window three tabs are present, the Map tab, the Detail tab, and the Printing Cart tab. The Detail tab displays the content of one or several abstracts by clicking either on the title or from the Printing Cart tab. Details are described below.


Related Abstract Search - search result for "neuroinformatics" - Mozilla Firefox
File Edit View History Bookmarks Tools Help
logo image
logo image
Show 10
Search
Related Abstract Search
Map
Detail
Printing cart
0.047
0.264
Done

neuroinformatics

© Lab for Neuroinformatics RIKEN, BSI

2.02096[s]

1. G: Novel Methods ... neuroinformat:2 >Related Abstracts >In Cart
☑ Finding hidden treasures: a related document search for SfN annual meeting abstracts
*S. USUI[1], N. L. KAMIJI[1], N. UEDA[2], T. TANIGUCHI[3]

1. G: Novel Methods ... >Related Abstracts >Add to Cart
□ Analyses over multiple scientific text corpora: combining MEDLINE, CRISP and the proceedings of the annual meetings of the Society for Neuroscience
*G. A. BURNS[1], D. NEWMAN[2], B. HERR[4], A. HOLLOWAY[3], P. SMYTH[2], K. BORNER[4]

2. G: Novel Methods ... >Related Abstracts >Add to Cart
□ Using CARMEN resources for high throughput multi-modal analysis and algorithm development
J. SIMONOTTO[1,2], S. EGLEN[4], C. ECHTERMEYER[5], C. ADAMS[2], *M. KAISER[3,2], E. SERNAGOR[2]

3. B: Neural ... >Related Abstracts >Add to Cart
□ Objective assessment and automated modeling of the different electrical classes of neurons in the neocortex
*S. DRUCKMANN[1], H. MARKRAM[2], I. SEGEV[1]

4. G: Novel Methods ... >Related Abstracts >Add to Cart
□ A literature mining and curation system for

| - | K01 | R01 | R02 | R03 | R04 | R05 | R06 | R07 | R08 | R09 | R10 | tf | df | tfidf | tf word | df word |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| K01 | K01 | 0.22 | 0.17 | 0.17 | 0.17 | 0.16 | 0.16 | 0.16 | 0.16 | 0.16 | 0.15 | - | - | - | - | - |
| R01 | 0.22 | R01 | 0.20 | 0.14 | 0.21 | 0.16 | 0.19 | 0.05 | 0.13 | 0.15 | 0.12 | - | - | - | - | - |
| R02 | 0.17 | 0.20 | R02 | 0.18 | 0.19 | 0.16 | 0.17 | 0.09 | 0.12 | 0.12 | 0.16 | - | - | - | - | - |
| R03 | 0.17 | 0.14 | 0.18 | R03 | 0.17 | 0.13 | 0.17 | 0.08 | 0.20 | 0.19 | 0.19 | - | - | - | - | - |
| R04 | 0.17 | 0.21 | 0.19 | 0.17 | R04 | 0.19 | [illegible] | 0.07 | 0.10 | 0.16 | 0.14 | - | - | - | - | - |
| R05 | 0.16 | 0.16 | 0.16 | 0.13 | 0.19 | R05 | [illegible] | 0.05 | 0.09 | 0.18 | 0.06 | - | - | - | - | - |
| R06 | 0.16 | 0.19 | 0.17 | 0.17 | [illegible] | [illegible] | R06 | 0.06 | 0.12 | 0.20 | 0.09 | - | - | - | - | - |
| R07 | 0.16 | 0.05 | 0.09 | 0.08 | 0.07 | 0.05 | 0.06 | R07 | 0.09 | 0.11 | 0.09 | - | - | - | - | - |
| R08 | 0.16 | 0.13 | 0.12 | 0.20 | 0.10 | 0.09 | 0.12 | 0.09 | R08 | 0.14 | 0.12 | - | - | - | - | - |
| R09 | 0.16 | 0.15 | 0.12 | 0.19 | 0.16 | 0.18 | 0.20 | 0.11 | 0.14 | R09 | 0.05 | - | - | - | - | - |
| R10 | 0.15 | 0.12 | 0.16 | 0.19 | 0.14 | 0.06 | 0.09 | 0.09 | 0.12 | 0.05 | R10 | - | - | - | - | - |

| - | K01 | R01 | R02 | R03 | R04 | R05 | R06 | R07 | R08 | R09 | R10 | tf | df | tfidf | tf word | df word |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| abstract | 12 | 5 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 19 | 182 | 82 | data | study |
| features | 2 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 28 | 756 | 81 | features | activity |
| mining | 0 | 1 | 0 | 0 | 8 | 0 | 1 | 0 | 0 | 1 | 0 | 11 | 22 | 71 | informat... | effects |
| spiking | 0 | 0 | 5 | 8 | 0 | 0 | 0 | 0 | 5 | 0 | 5 | 23 | 812 | 65 | spiking | neurons |
| tool | 5 | 1 | 3 | 1 | 2 | 5 | 1 | 0 | 0 | 0 | 0 | 18 | 481 | 60 | neurons | increasi... |
| informat... | 2 | 1 | 2 | 2 | 5 | 0 | 1 | 0 | 4 | 1 | 9 | 27 | 1482 | 60 | abstract | function... |
| search | 6 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 3 | 2 | 13 | 209 | 54 | tool | brain |
| database | 2 | 2 | 0 | 1 | 1 | 0 | 2 | 0 | 0 | 2 | 0 | 10 | 73 | 52 | analysis | present |
| neurosci... | 3 | 2 | 1 | 1 | 3 | 1 | 2 | 0 | 1 | 2 | 0 | 16 | 647 | 49 | neurosci... | modeling |
| extracte... | 5 | 0 | 2 | 1 | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 13 | 428 | 45 | process... | data |
| scientif... | 0 | 5 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 8 | 52 | 44 | based | times |

**Fig. 3.** The Related Abstract Search Tool. The left side shows a list of abstracts with key abstracts in yellow background and related abstracts in blue background. The right shows the similarity map, and bellow the map the list of extracted words with the number of occurrence on each abstract.

### 3.1 Related Abstract

Related abstracts are calculated online, since it also allow multiple selected key abstracts. Assuming each abstract as a vector with each element being the words extracted from all abstracts, the similarity is calculated as the cosine between vectors. When calculating the related abstract of multiple selected abstracts, a composite vector is first calculated from the selected key abstracts, and then the related abstract is calculated assuming the composite vector as a single abstract. Related abstracts that did not contain any of the keywords entered for searching the key abstracts are marked with a gold ingot meaning it is a hidden treasure. That is, abstracts that may be related to the topic that the user is searching for, but without any of the keywords the user has chosen.

### 3.2 Similarity Map

The similarity map is a color map displayed in the Map tab (Fig 3 right side). The colors represents relatively the highest (red) and lowest (blue) similarity between two abstracts. This feature was implemented to help users in easily identifying abstracts belonging to the same group, or abstracts independent to others. That is, abstracts showing higher similarity index has higher possibility in being discussing about similar topics, as well as, abstracts with low similarity index may belong to a different research field. The first abstracts starting with "K" are for the key abstracts, with the number indicating the order in the list. Similarly, the following abstracts starting with "R" are for the related abstracts. By clicking on one of the colors over the map, the two abstracts in question are displayed in a pop-up window for easy comparison of their titles. Utilizing this feature, we have found that abstracts with a much higher similarity compared to others are often from the same research group. Moreover, another interesting information this map may provide is, when there are two or more abstracts with high similarity in a row or column, a user may be also interested in knowing whether these abstracts has also high similarity to each other. That is, Suppose R03 and R07 are very similar to K01. If the similarity between R03 and R07 is also high, this may indicate that both abstracts might be of interest if one of them is. However, if the similarity between R03 and R07 is low, it may indicate that both works points to a different direction in a given topic. Similar to this, if one of the Related abstracts has very low similarity to all other abstracts, this may indicate that this given abstract is similar to the key abstract but again pointing to a different direction compared to all other abstracts. This may provide user with rich information in determining which related abstracts to look at.

Below the table, a list of words is displayed according to the term frequency- inverse document frequency (tf-idf) index. Following the words, a circle indicates the occurrence of that word in a given abstract (rows). This list is displayed in order to help users in narrowing their result and also to suggest other keywords that may be utilized to refine their search. Moreover, by clicking on a term, it is automatically added to the search box.

### 3.3 Printing Cart

The printing cart stores selected abstracts for later reviewing or printing the list. Abstracts are added to the printing cart by clicking on the “Add to Cart” link at each abstract title. Clicking on the Add to Cart on the pop up window displayed at the similarity map also has the same effect. Abstracts added to the printing cart are sorted according to its presentation time slot. On top of the printing cart, three buttons are present; “Show Detail”, “Print Preview” and “Print”. These buttons take effect only to the abstracts whose check boxes are turned on. The Show Detail button displays the content of the selected abstracts on the Detail tab. The Print Preview displays the content of the selected abstract on a new window, so users can check prior to printing the list of abstracts. And finally, the Print button displays the print dialog box for printing the selected abstracts.

## 4 Discussions and Perspectives

The Related Abstract Search Tool, inspired on the 3D-SE viewer, aims to help not only searching abstracts submitted to scientific conferences, but it may also provide a tool for searching and browsing through any type of database. By graphically displaying the relationship between items, the task of searching becomes easier as more information is provided at a glance.

When dealing with documents, choosing the correct keywords becomes essential. As the present work deals with abstracts submitted to the Society for Neuroscience annual meeting, we have filtered the extracted words with the keyword list provided on the last issue of the Journal of Neuroscience 2006 and 2007. Not so much surprising, filtering have worsen the output mainly because keywords necessary for newly emerging field such as neuroinformatics are not included. We have not analyzed the extent of missing keywords proposed by the Journal of Neuroscience since it is not our scope in this work. However, this result suggests that part of this tool may be utilized to help journal editors in reviewing their keyword list by providing possible keyword candidates.

To further support not only abstract searching, a feature for linking the results stored into the printing cart with schedule manager such as iCal, google calendar and others is also being developed. Moreover, based on the abstracts the user has added to the printing cart, we aim to provide users with other possible abstracts he/she might get interested to which are hardly found only by the keyword and related abstract search features discussed above.

## References

1. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM 18, 613–620 (1975)
2. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
3. Börner, K., Chen, C., Boyack, K.W.: Visualizing Knowledge Domains. ARIST 37, 179–255 (2003)
4. Iwata, T., Saito, K., Ueda, N.: Parametric Embedding for Class Visualization. Neural Comp. 19, 2536–2556 (2007)
5. Saito, K., Iwata, T., Ueda, N.: Visualization of Bipartite Graph by Spherical Embedding. In: 14th Japanese Neural Network Society (2004)
6. Usui, S., Naud, A., Ueda, N., Taniguchi, T.: 3D-SE Viewer: A Text Mining Tool based on Bipartite Graph Visualization. In: 20th International Joint Conference on Neural Networks, pp. 1103–1108 (2007)
7. Naud, A.P., Usui, S., Ueda, N., Taniguchi, T.: Visualization of documents and concepts in neuroinformatics with the 3D-SE viewer. Front. Neuroinform. 1, 7 (2007)
8. Usui, S., Kamiji, N.L., Ueda, N., Taniguchi, T.: Finding hidden treasures: a related document search for SfN annual meeting abstracts. SfN 2008 Washington DC, 396.3/UU61 (2008)
9. Usui, S., Kamiji, N.L., Okumura, Y., Taniguchi, T., Ueda, N.: Databasing and Visualizing Neuroscience Knowledge: XooNIps and 3D-SE Viewer. In: 15th International Conference on Neural Information Processing, ICONIP (2008)
10. BSI Team Map, http://www.brain.riken.jp/en/teammap/
11. Related Abstract Search(beta) for SfN 2008 (2008), http://ras.ni.brain.riken.jp
12. Usui, S., Palmes, P., Nagata, K., Taniguchi, T., Ueda, N.: Keyword extraction, ranking, and organization for the neuroinformatics platform. Biosystems 88, 334–342 (2007)

# An Artificial Bee Colony Algorithm for the Quadratic Knapsack Problem

Srikanth Pulikanti and Alok Singh

Department of Computer and Information Sciences
University of Hyderabad, Hyderabad – 500046, India
srikanthpulikanti@yahoo.in, alokcs@uohyd.ernet.in

**Abstract.** In this paper we have proposed a new hybrid approach combining artificial bee colony algorithm with a greedy heuristic and a local search for the quadratic knapsack problem. Quadratic knapsack problem belongs to traditional knapsack problem family and it is an extension of the well-known 0/1 knapsack problem. In this problem profits are also associated with pairs of objects along with individual objects. As this problem is an extension of the 0/1 knapsack problem, it is also $\mathcal{NP}$-Hard. Artificial bee colony algorithm is a new swarm intelligence technique inspired by foraging behavior of natural honey bee swarms. Performance of our algorithm on standard quadratic knapsack problem instances is compared with the other best heuristic techniques. Results obtained on these instances show that our hybrid artificial bee colony algorithm is superior to these techniques in many aspects.

**Keywords:** Artificial bee colony algorithm, combinatorial optimization, quadratic knapsack problem, swarm intelligence.

## 1 Introduction

In traditional 0/1 knapsack problem for a given knapsack with fixed capacity $C$ and a given set of $n$ objects with positive weights and positive profit values, we have to select a subset of these $n$ objects for inclusion in the knapsack in such a way that the total weights of the selected objects should not exceed its capacity $C$, and the total profit value $P$ of selected objects should be as high as possible. If the presence of an object $i$ in the knapsack is represented by a binary variable $o_i$, whose value will be 1 if the object $i$ is present in the knapsack, otherwise it will be 0, then the 0/1 knapsack problem can be mathematically formulated as follows: Maximize $P = \sum_{i=1}^{n} o_i p_i$, subject to $\sum_{i=1}^{n} o_i w_i \leq C$, where $w_i$ and $p_i$ are respectively the weight and profit of object $i$.

Whereas in 0/1 knapsack problem profit values are associated only with individual objects, in quadratic knapsack problem additional non-negative profit values are associated with every pair of objects. The profit value associated with a pair of objects is added to knapsack's total profit value if and only if both the objects of the pair are present in the knapsack. If both object $i$ and object $j$ is present in the knapsack, then a profit value associated with this pair $p_{ij}$ is

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 196–205, 2009.

added to the total knapsack profit value, along with $i$ and $j$ individual profit values. Thus, quadratic knapsack problem can be formulated as follows: Maximize $P = \sum_{i=1}^{n} o_i p_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} o_i o_j p_{ij}$, subject to $\sum_{i=1}^{n} o_i w_i \leq C$. This problem can be reduced to traditional 0/1 knapsack problem, by setting all the profit values associated with pairs of objects to zero. Thus, being a generalization of the 0/1 knapsack problem, this problem is also $\mathcal{NP}$-Hard.

In this paper we have used the concept of absolute value density and relative value density which were first defined in [1]. The absolute value density of an object is the the sum of all the profit values associated with an object divided by its weight. The relative value density of an object is defined relative to a subset of objects. The relative value density of an object with respect to a subset of objects is equal to its absolute value density in the QKP instance obtained by considering only the objects belonging to the subset and the object in question.

Quadratic knapsack problem (QKP) was introduced by Gallo *et al.* [2] where a branch-and-bound algorithm was proposed for solving this problem. Since then many exact methods for QKP have been developed. The reader is referred to [3] for a survey of these methods as well as for practical applications of QKP. Among the heuristic techniques, Julstrom [1] proposed three methods for QKP - a greedy technique, a genetic algorithms and a greedy genetic algorithms. The genetic and greedy genetic algorithms differ in that the former is a normal genetic algorithm but the latter contains implementation of a greedy heuristic in crossover and mutation operators. A hybrid technique combining steady state genetic algorithm with a greedy heuristic was developed by Singh and Baghel [4]. This hybrid approach was tested on the same instances as used in [1], and, it outperformed all the three approaches of [1]. An algorithm based on swarm intelligence was proposed by Xie and Liu [5]. They used a new swarm-based technique called mini-swarm, which was developed from MAOS [6]. Mini-swarm can be considered as a group or society of agents, which are individual in nature and having a limited amount of longterm memory. These agents cooperate with each other to find a optimal solution for a given problem. This mini-swarm approach also outperformed the best approach of [1].

In this paper we have proposed a hybrid swarm intelligence technique combining an artificial bee colony (ABC) algorithm with a greedy heuristic and a local search for solving the quadratic knapsack problem.

This paper is organized as follows: Section 2 provides an introduction to artificial bee colony algorithm, section 3 describes our hybrid swarm intelligence approach. Computation results are presented in section 4, whereas section 5 outlines conclusions and future work.

## 2 Artificial Bee Colony Algorithm

Artificial bee colony algorithm is a new metaheuristic technique inspired by the foraging behaviour of natural honey bee swarms. This technique was proposed by Karaboga [7] and improved further by Karaboga and Basturk [8] and Singh [9]. In natural honey bee colonies, foraging bees are classified into three types, viz., employed bees, onlooker bees and scout bees. All bees which are presently exploiting a food source are termed as employed. The employed bees bring loads of nectar from the food source to the hive and may share the information about their food sources with onlooker bees. Onlookers are those bees which are waiting in the hive for the employed bees to share information about the food sources presently being exploited by them. Scouts are those bees that are presently searching for new food sources in the neighborhood of the hive. Employed bees share information about their food sources by dancing in a common area in the hive called dance area. The nature and the duration of a dance is proportional to the nectar content of the food source presently being tapped by the dancing bee. Onlooker bees observe numerous dances before choosing a food source. The onlookers have a tendency to choose a food source with a probability proportional to the nectar content of that food source. Therefore, good food sources attract more bees than the bad ones. Whenever a bee, whether it is scout or onlooker, finds a food source it becomes employed. Whenever a food source is completely exhausted, all the employed bees associated with it leave it, and can again become scouts or onlookers.

Motivated by this natural phenomenon, Karaboga [7] proposed the artificial bee colony(ABC) algorithm. In ABC algorithm each food source represents a feasible solution to the problem under consideration and the nectar content of a food source represents the fitness of the solution represented by that food source. Like the natural bee colony, in artificial bee colony also artificial bees are classified into same three types with similar functions, viz., employed bees, onlooker bees and scout bees. In ABC algorithm, there are only as many employed bees as there are number of food sources. Usually, the number of onlookers are also taken to be equal to the number of employed bees. The employed bee of an exhausted food source becomes a scout and as soon as it finds a new food source it again becomes employed.

ABC algorithm follows an iterative process. It starts by associating each employed bee with one randomly generated food source (solution).Then during each iteration, each employed bee determines a food source in the neighborhood of its currently associated food source and evaluates its nectar content (fitness). If its nectar content is better in comparison to that of its currently associated food source then that employed bee moves to this new food source abandoning the old one, otherwise, it remains at its old food source. After all employed bees have completed this process, they share the nectar information of the food sources with the onlookers. Based on this information, each onlooker select a food source according to the probability proportional to the nectar content of that food source. The probability $p_i$, with which a food source $i$ is selected by an onlooker bee is given by $p_i = f_i / \sum_{j=1}^{ns} f_j$, where $f_i$ is the fitness of the solution $i$ and $ns$ is the total number of food sources. This selection scheme is known by the name ”roulette wheel selection” among the genetic algorithm community. Due to this selection scheme, it is assured that solutions with higher fitness values are selected more often than the solutions with lower fitness values. Once all onlookers have selected a food source, each of them determines a food source in the neighborhood of its selected food source and computes the fitness value of this newly determined food source. The best food source, among all the food sources determined by the onlookers associated with a particular food source $i$, and, the food source $i$ itself, will be the new location of the food source $i$ in the next iteration. If the fitness of a solution associated with a particular food source does not improve for a predetermined number of iterations then the employed bee associated with it abandons it to become a scout. This scout is transformed into an employed bee immediately by associating it with a randomly generated new solution. Once new location of each food source is determined, the next iteration of the ABC algorithm begins. This process is repeated over and over again as long as the termination condition remain unsatisfied.

## 3 The Hybrid ABC Algorithm for QKP

We have developed a hybrid approach combining ABC algorithm with a heuristic and a local search. The heuristic transforms the infeasible solution obtained through ABC algorithm into a good feasible solution which is further improved by local search. The heuristic and local search are discussed in subsection 3.4. The main features of our ABC algorithm for the QKP problem are described in subsequent subsections.

### 3.1 Solution Encoding

Similar to [1,4], we have also used a bit-vector of length $n$ to represent a solution where $n$ is the total number of objects. If the value at the $i^{th}$ position of the bit-vector is 1 then it indicates that object $i$ is in the solution, whereas a 0 at the $i^{th}$ position indicates that object $i$ is not in the solution.

### 3.2 Generation of Initial Solutions

We have used the same method as used in [1] for generating initial solutions. We have also used this method for generating a new solution whenever a bee becomes a scout. This method generates a solution by following an iterative process. During each iteration, this method selects two objects randomly from the set of unassigned objects and performs a binary tournament between them. With probability $p_{init}$, object with higher absolute value density will be selected and with probability $(1 - p_{init})$ object with lower absolute value density will be selected. The selected object will be deleted from the list of unassigned objects, and, if it fits, it is included in the knapsack. This process is repeated till the set of unassigned objects becomes empty.

### 3.3 Probability of Selecting a Food Source

Instead of selecting the food source for an onlooker through roulette wheel selection, we have used binary tournament selection. In binary tournament selection, two food sources are randomly chosen and better of the two food sources are selected with probability $p_{onl}$ and worse of the two with probability $(1 - p_{onl})$.We have tried the roulette wheel selection also, but binary tournament selection always gave better results.

### 3.4 Determination of a New Neighboring Food Source

In order to generate a solution in the neighborhood of a particular solution say solution $i$, first we create a copy $i'$ of solution $i$ and choose another solution randomly. From this randomly chosen solution we add a maximum of 5 objects, randomly chosen from those objects not present in $i'$, to $i'$. If we are not able to find even one such object then that means that the two solutions are identical. This situation was called collision in [9] and is resolved in the same manner as in [9]. If this collision occurs while generating a neighboring solution for an employed bee then it is resolved by making the associated employed bee scout thereby eliminating one duplicate solution. If the collision occurs while generating the neighboring solution for an onlooker then another solution is chosen randomly for copying objects from it. This process is repeated till we found a solution different from $i'$.

After some objects are added to $i'$, it becomes infeasible, i.e., violates the knapsack capacity constraint. To transform this infeasible solution into a good feasible solution we have used a slightly modified version of the two-phase heuristic used in [4]. The first phase of the heuristic transforms the infeasible solution into a feasible one. It begins by computing the relative value densities of every object present in the solution with respect to the set of all objects present in solution. Based on these relative value densities, next a binary tournament is performed between the two randomly chosen objects of the solution. With probability $p_{tour}$, the object with smaller relative value density is deleted from the solution and with probability $(1 - p_{tour})$ the object with larger relative value density. The relative value densities of remaining objects are updated accordingly. Next the heuristic iteratively deletes objects from the chromosome until the chromosome becomes feasible. During each iteration the objects with smallest relative value density is chosen for deletion and the relative value densities of remaining objects are updated accordingly. The second phase iteratively tries to include as many unassigned objects as it can to the feasible solution using the same criteria as used in relative greed heuristic of [1]. It begins by computing the relative value density of every unassigned object with respect to the set of objects present in the solution. During each iteration it adds to the solution, the unassigned object with highest relative value density whose addition does not cause the capacity violation and the relative value densities of the unassigned objects are updated accordingly. This process is repeated until it is not possible to add any more objects.

Once the heuristic phase is over, in some cases we try to improve the solution further using a local search. The local search is based on repeatedly exchanging an object in the knapsack with an unassigned object. We exchange an object in the knapsack with an unassigned object, if such an exchange can increase the overall profit without violating the capacity constraint. All objects present in the knapsack are tried one-by-one in their natural order for exchange with unassigned objects. As soon as we find a profitable exchange for an object in the knapsack, we immediately perform this exchange. This local search is used with probability $p_{exch}$. Actually, we found that using the local search always not only increases the execution times but also sometimes slighty decreases the solution quality. The latter can be attributed to lack of diversity resulting from using the local search always.

### 3.5 Other Features

Unlike the usual practice, we have used different number of employed bees and onlooker bees. If the solution associated with an employed bee does not improve for *itnoimp* number of iterations then it becomes a scout. Like [9], here also there is a second possibility in which an employed bee can become scout as described previously. Also, there is no upper limit on the number of scouts in a single iteration. The number of scouts in a particular iteration depends on the number of times the aforementioned two conditions hold.

## 4 Results and Discussions

We have implemented our approach in C and executed it on an Intel core 2 duo system with 2 GB RAM running at 3.0 GHz under OpenSUSE 11.1. We have executed our program on a set of 80 QKP benchmark instances of Billionet and Soutif [10]. Out of 80 instances 40 instances have $n = 100$ objects, and other 40 instances have $n = 200$ objects. These instances are further subdivided according to their densities (the percentage of non-zero profit values associated with pairs of objects). For each value of density $\in \{0.25, 0.5, 0.75, 1.0\}$ there are 10 instances for each of the two values of $n$. This leads to a total of 80 benchmark instances. Optimal solution values of all of these instances are known [10,11].

We have used a colony of 200 bees, out of which 50 are employed bees and 150 are onlookers. Our algorithm terminates when the best solution does not improve over 500 iteration. We have used $itnoimp = 2n, p_{init} = 0.75, p_{tour} = 0.8, p_{onl} = 0.95$ and $p_{exch} = 0.1$. These parameter values are chosen empirically.

We have compared our hybrid ABC approach with mini-swarm approach of [5] and HSSGA of [4] which are the best heuristics for the problem. Mini-swarm approach was executed 100 times on each of the 80 instances, whereas HSSGA was executed 50 times on each of the selected 20 instances(10 instances with $n = 100$, density =0.25 and 10 instances with $n = 200$, density =1.0). Like mini-swarm, our algorithm is also executed 100 times on each instance. Table 1 compares the performance of our approach with mini-swarm on instances with $n = 100$, whereas table 2 does the same for $n = 200$. For each instance these tables report for each method the number of times the optimal solution is found out of 100 trials($Hits$), the highest profit solution found by each method($Max$) and average time to reach the best solution($AvgT$). For our approach these

**Table 1.** Comparision of Mini-swarm and ABC on instances with 100 objects

| **Instance** | | | **Mini-swarm** | | | **ABC** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Time | | | | | Time | |
| d | num | Optval | Hits | Max | AvgT | Hits | Max | Avg | SD | MinT | AvgT |
| 0.25 | 1 | 18558 | 100 | 18558 | 0.430 | 100 | 18558 | 18558 | 0.0 | 0.02 | 0.039 |
| | 2 | 56525 | 100 | 56525 | 0.170 | 100 | 56525 | 56525 | 0.0 | 0.00 | 0.017 |
| | 3 | 3752 | 100 | 3752 | 0.190 | 100 | 3752 | 3752 | 0.0 | 0.00 | 0.011 |
| | 4 | 50382 | 100 | 50382 | 0.920 | 100 | 50382 | 50382 | 0.0 | 0.01 | 0.018 |
| | 5 | 61494 | 100 | 61494 | 0.064 | 100 | 61494 | 61494 | 0.0 | 0.00 | 0.008 |
| | 6 | 36360 | 100 | 36360 | 0.415 | 100 | 36360 | 36360 | 0.0 | 0.02 | 0.024 |
| | 7 | 14567 | 100 | 14657 | 0.233 | 100 | 14657 | 14657 | 0.0 | 0.01 | 0.019 |
| | 8 | 20452 | 100 | 20452 | 0.313 | 100 | 20452 | 20452 | 0.0 | 0.02 | 0.023 |
| | 9 | 35438 | 39 | 35438 | 0.801 | 100 | 35438 | 35438 | 0.0 | 0.02 | 0.040 |
| | 10 | 24930 | 100 | 24930 | 0.890 | 100 | 24930 | 24930 | 0.0 | 0.02 | 0.037 |
| 0.50 | 1 | 83742 | 100 | 83742 | 0.273 | 100 | 83742 | 83742 | 0.0 | 0.01 | 0.032 |
| | 2 | 104856 | 52 | 104856 | 0.224 | 100 | 104856 | 104856 | 0.0 | 0.00 | 0.020 |
| | 3 | 34006 | 88 | 34006 | 0.366 | 100 | 34006 | 34006 | 0.0 | 0.01 | 0.023 |
| | 4 | 105996 | 100 | 105996 | 0.137 | 100 | 105996 | 105996 | 0.0 | 0.00 | 0.012 |
| | 5 | 56464 | 100 | 56464 | 0.669 | 100 | 56464 | 56464 | 0.0 | 0.02 | 0.031 |
| | 6 | 16083 | 100 | 16083 | 0.263 | 100 | 16083 | 16083 | 0.0 | 0.01 | 0.022 |
| | 7 | 52819 | 100 | 52819 | 0.417 | 100 | 52819 | 52819 | 0.0 | 0.01 | 0.028 |
| | 8 | 54246 | 100 | 54246 | 0.416 | 100 | 54246 | 54246 | 0.0 | 0.01 | 0.027 |
| | 9 | 68974 | 100 | 68974 | 0.950 | 100 | 68974 | 68974 | 0.0 | 0.01 | 0.025 |
| | 10 | 88634 | 100 | 88634 | 0.345 | 100 | 88634 | 88634 | 0.0 | 0.01 | 0.036 |
| 0.75 | 1 | 189137 | 100 | 189137 | 0.090 | 100 | 189137 | 189137 | 0.0 | 0.00 | 0.002 |
| | 2 | 95074 | 75 | 95074 | 0.749 | 100 | 95074 | 95074 | 0.0 | 0.02 | 0.075 |
| | 3 | 62098 | 100 | 62098 | 0.349 | 100 | 62098 | 62098 | 0.0 | 0.02 | 0.025 |
| | 4 | 72245 | 100 | 72245 | 0.522 | 100 | 72245 | 72245 | 0.0 | 0.02 | 0.072 |
| | 5 | 27616 | 100 | 27616 | 0.210 | 100 | 27616 | 27616 | 0.0 | 0.01 | 0.018 |
| | 6 | 145273 | 100 | 145273 | 0.555 | 100 | 145273 | 145273 | 0.0 | 0.01 | 0.016 |
| | 7 | 110979 | 100 | 110979 | 0.341 | 100 | 110979 | 110979 | 0.0 | 0.01 | 0.029 |
| | 8 | 19570 | 100 | 19570 | 0.192 | 100 | 19570 | 19570 | 0.0 | 0.01 | 0.016 |
| | 9 | 104341 | 100 | 104341 | 0.417 | 100 | 104341 | 104341 | 0.0 | 0.02 | 0.044 |
| | 10 | 143740 | 100 | 143740 | 0.209 | 100 | 143740 | 143740 | 0.0 | 0.01 | 0.016 |
| 1.00 | 1 | 81978 | 100 | 81978 | 0.271 | 100 | 81978 | 81978 | 0.0 | 0.02 | 0.036 |
| | 2 | 190424 | 100 | 190424 | 0.230 | 100 | 190424 | 190424 | 0.0 | 0.01 | 0.019 |
| | 3 | 225434 | 100 | 225434 | 0.190 | 100 | 225434 | 225434 | 0.0 | 0.00 | 0.011 |
| | 4 | 63028 | 100 | 63028 | 0.400 | 100 | 63028 | 63028 | 0.0 | 0.01 | 0.022 |
| | 5 | 230076 | 100 | 230076 | 0.106 | 100 | 230076 | 230076 | 0.0 | 0.00 | 0.009 |
| | 6 | 74358 | 100 | 74358 | 0.330 | 100 | 74358 | 74358 | 0.0 | 0.01 | 0.032 |
| | 7 | 10330 | 100 | 10330 | 0.120 | 100 | 10330 | 10330 | 0.0 | 0.00 | 0.007 |
| | 8 | 62582 | 100 | 62582 | 0.257 | 100 | 62582 | 62582 | 0.0 | 0.01 | 0.023 |
| | 9 | 232754 | 100 | 232754 | 0.109 | 100 | 232754 | 232754 | 0.0 | 0.00 | 0.028 |
| | 10 | 193262 | 100 | 193262 | 0.240 | 100 | 193262 | 193262 | 0.0 | 0.01 | 0.123 |
| | | **Total hits** | 3854/4000 | | | 4000/4000 | | | | | |

**Table 2.** Comparison of Mini-swarm and ABC on instances with 200 objects

| **Instance** | | | **Mini-swarm** | | | **ABC** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Time | | | | | Time | |
| d | num | Optval | Hits | Max | AvgT | Hits | Max | Avg | SD | MinT | AvgT |
| 0.25 | 1 | 204441 | 100 | 204441 | 1.221 | 100 | 204441 | 204441 | 0.0 | 0.08 | 0.118 |
| | 2 | 239573 | 100 | 239573 | 0.499 | 100 | 239573 | 239573 | 0.0 | 0.03 | 0.044 |
| | 3 | 245463 | 98 | 245463 | 0.536 | 100 | 245463 | 245463 | 0.0 | 0.00 | 0.047 |
| | 4 | 222361 | 100 | 222361 | 0.726 | 100 | 222361 | 222361 | 0.0 | 0.07 | 0.141 |
| | 5 | 187324 | 100 | 187324 | 1.571 | 100 | 187324 | 187324 | 0.0 | 0.12 | 0.207 |
| | 6 | 80351 | 28 | 80351 | 2.781 | 100 | 80351 | 80351 | 0.0 | 0.25 | 0.341 |
| | 7 | 59036 | 77 | 59036 | 2.583 | 100 | 59036 | 59036 | 0.0 | 0.14 | 0.286 |
| | 8 | 149433 | 100 | 149433 | 2.038 | 100 | 149433 | 149433 | 0.0 | 0.22 | 0.360 |
| | 9 | 49366 | 100 | 49366 | 1.405 | 100 | 49366 | 49366 | 0.0 | 0.12 | 0.150 |
| | 10 | 48459 | 100 | 48459 | 0.940 | 100 | 48459 | 48459 | 0.0 | 0.15 | 0.201 |
| 0.50 | 1 | 372097 | 100 | 372097 | 1.292 | 100 | 372097 | 372097 | 0.0 | 0.15 | 0.201 |
| | 2 | 211130 | 26 | 211130 | 1.825 | 57 | 211130 | 211121.4 | 9.9 | 0.36 | 5.073 |
| | 3 | 227185 | 100 | 227185 | 1.606 | 100 | 227185 | 227185 | 0.0 | 0.26 | 0.344 |
| | 4 | 228572 | 100 | 228572 | 4.801 | 100 | 228572 | 228572 | 0.0 | 0.15 | 0.200 |
| | 5 | 479651 | 100 | 479651 | 0.501 | 100 | 479651 | 479651 | 0.0 | 0.00 | 0.047 |
| | 6 | 426771 | 100 | 426771 | 1.053 | 100 | 426771 | 426771 | 0.0 | 0.04 | 0.084 |
| | 7 | 220890 | 100 | 220890 | 2.066 | 100 | 220890 | 220890 | 0.0 | 0.28 | 0.339 |
| | 8 | 317952 | 100 | 317952 | 1.726 | 100 | 317952 | 317952 | 0.0 | 0.15 | 0.196 |
| | 9 | 104936 | 100 | 104936 | 1.097 | 100 | 104936 | 104936 | 0.0 | 0.13 | 0.167 |
| | 10 | 284751 | 98 | 284751 | 2.097 | 100 | 284751 | 284751 | 0.0 | 0.16 | 0.286 |
| 0.75 | 1 | 442894 | 16 | 442894 | 3.339 | 35 | 442894 | 442758.5 | 101.8 | 0.24 | 7.096 |
| | 2 | 286643 | 95 | 286643 | 2.797 | 100 | 286643 | 286643 | 0.0 | 0.20 | 0.495 |
| | 3 | 61924 | 100 | 61924 | 0.656 | 100 | 61924 | 61924 | 0.0 | 0.07 | 0.103 |
| | 4 | 128351 | 100 | 128351 | 1.232 | 100 | 128351 | 128351 | 0.0 | 0.17 | 0.204 |
| | 5 | 137885 | 100 | 137885 | 1.127 | 100 | 137885 | 137885 | 0.0 | 0.09 | 0.125 |
| | 6 | 229631 | 100 | 229631 | 1.661 | 100 | 229631 | 229631 | 0.0 | 0.25 | 0.301 |
| | 7 | 269887 | 100 | 269887 | 6.291 | 100 | 269887 | 269887 | 0.0 | 0.24 | 0.351 |
| | 8 | 600858 | 98 | 600858 | 1.966 | 100 | 600858 | 600858 | 0.0 | 0.08 | 0.120 |
| | 9 | 516771 | 100 | 516771 | 1.515 | 100 | 516771 | 516771 | 0.0 | 0.16 | 0.284 |
| | 10 | 14269 | 100 | 142694 | 1.072 | 100 | 142694 | 142694 | 0.0 | 0.17 | 0.194 |
| 1.00 | 1 | 937149 | 100 | 937149 | 0.574 | 100 | 937149 | 937149 | 0.0 | 0.00 | 0.043 |
| | 2 | 303058 | 100 | 303058 | 1.179 | 100 | 303058 | 303058 | 0.0 | 0.29 | 0.643 |
| | 3 | 29367 | 100 | 29367 | 0.301 | 100 | 29367 | 29367 | 0.0 | 0.02 | 0.039 |
| | 4 | 100838 | 100 | 100838 | 0.543 | 100 | 100838 | 100838 | 0.0 | 0.10 | 0.129 |
| | 5 | 7866355 | 100 | 786635 | 2.043 | 100 | 786635 | 786635 | 0.0 | 0.08 | 0.127 |
| | 6 | 41171 | 100 | 41171 | 0.311 | 100 | 41171 | 41171 | 0.0 | 0.04 | 0.062 |
| | 7 | 701094 | 100 | 701094 | 1.732 | 100 | 701094 | 701094 | 0.0 | 0.16 | 0.340 |
| | 8 | 782443 | 100 | 782443 | 1.258 | 100 | 782443 | 782443 | 0.0 | 0.12 | 1.477 |
| | 9 | 628992 | 100 | 628992 | 2.277 | 100 | 628992 | 628992 | 0.0 | 0.16 | 0.984 |
| | 10 | 378442 | 100 | 378442 | 1.757 | 100 | 378442 | 378442 | 0.0 | 0.29 | 0.350 |
| | | **Total hits** | 3736/4000 | | | 3892/4000 | | | | | |

**Table 3.** Performance of HSSGA on 20 instances

| **Instance** | | | | **HSSGA** | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Time | |
| n | d | num | Optval | Hits | Max | Avg | SD | MinT | AvgT |
| 100 | 0.25 | 1 | 18558 | 50 | 18558 | 18558 | 0.0 | 0.00 | 0.03 |
| | | 2 | 56525 | 50 | 56525 | 56525 | 0.0 | 0.00 | 0.01 |
| | | 3 | 3752 | 50 | 3752 | 3752 | 0.0 | 0.00 | 0.01 |
| | | 4 | 50382 | 50 | 50382 | 50382 | 0.0 | 0.00 | 0.01 |
| | | 5 | 61494 | 50 | 61494 | 61494 | 0.0 | 0.00 | 0.01 |
| | | 6 | 36360 | 50 | 36360 | 36360 | 0.0 | 0.00 | 0.01 |
| | | 7 | 14657 | 50 | 16457 | 14657 | 0.0 | 0.00 | 0.01 |
| | | 8 | 20452 | 50 | 20452 | 20452 | 0.0 | 0.00 | 0.01 |
| | | 9 | 35438 | 50 | 35438 | 35438 | 0.0 | 0.00 | 0.01 |
| | | 10 | 24930 | 50 | 24930 | 24930 | 0.0 | 0.00 | 0.01 |
| | | | **Total hits** | 500/500 | | | | | |
| 200 | 1.0 | 1 | 937149 | 50 | 937149 | 937149 | 0.0 | 0.39 | 0.97 |
| | | 2 | 303058 | 50 | 303058 | 303058 | 0.0 | 0.60 | 8.50 |
| | | 3 | 29367 | 50 | 29367 | 29367 | 0.0 | 0.01 | 0.02 |
| | | 4 | 100838 | 50 | 100838 | 100838 | 0.0 | 0.02 | 0.07 |
| | | 5 | 786635 | 50 | 786635 | 786635 | 0.0 | 1.77 | 7.56 |
| | | 6 | 41171 | 50 | 41171 | 41171 | 0.0 | 0.01 | 0.03 |
| | | 7 | 701094 | 50 | 701094 | 701094 | 0.0 | 0.76 | 12.26 |
| | | 8 | 782443 | 50 | 782443 | 782443 | 0.0 | 21.54 | 54.28 |
| | | 9 | 628992 | 50 | 628992 | 628992 | 0.0 | 0.26 | 5.39 |
| | | 10 | 378442 | 50 | 378442 | 378442 | 0.0 | 0.05 | 0.46 |
| | | | **Total hits** | 500/500 | | | | | |

tables also report the average solution quality, the standard deviation of solution values and the minimum time to reach the best solution. Data for mini-swarm is taken from [5] where average solution quality and the minimum time to reach the best solution is not reported. These tables clearly shows the superiority of our approach over mini-swarm in terms of solution quality. For 100 objects instances our approach finds optimal solution in all 4000 trials, whereas mini-swarm approach gives optimal solutions in 3854 trials out of 4000 trials. For 200 objects instances our approach finds optimal solution in 3892 trials, whereas mini-swarm finds optimal solution in 3736 trials. Overall, there are only two instances on which hit rate for our method is less than 100%, whereas for mini-swarm there are 12 instances on which hit rate is less than 100%. The mini-swarm approach was executed on a Pentium 4 system running at 3.06GHz under Red Hat Linux 7.3. As it is executed on a different system than our approach, therefore, it is not possible to exactly compare the time of two approaches. However, we can safely say that our approach is faster on most of the instances even after compensating for the processing speeds.

Table 3 reports the result of HSSGA on 20 instances. Data for HSSGA is taken from [4]. Comparing table 3 with tables 1 and 2, we can see that our approach as well as HSSGA have 100% hit rate on all the 20 instances. HSSGA was executed on a Pentium 4 system running at 2.4GHz under Red Hat Linux 9.0. Though it is not possible to exactly compare the speed of the two approaches, it can be easily inferred from the tables that HSSGA is faster on all instances with $n = 100$, whereas our hybrid ABC approach is faster on most of the instances with $n = 200$.

## 5 Conclusions

We have developed a hybrid artificial bee colony approach for the quadratic knapsack problem and compared it with mini-swarm [5] and HSSGA [4], which are the best heuristics known for the problem. Our approach outperformed the mini-swarm approach both in terms of solution quality as well as running time, whereas its performance is comparable to HSSGA. The superiority over mini-swarm approach is significant as it is another swarm-based approach. As a future work we plan to extend our approach to quadratic multiple knapsack problem.

## References

1. Julstrom, B.A.: Greedy, Genetic, and Greedy Genetic Algorithms for the Quadratic Knapsack Problem. In: Proceedings of the 2005 Genetic and Evolutionary Computation Conference, pp. 607–614. ACM Press, New York (2005)
2. Gallo, G., Hammer, P.L., Simeone, B.: Quadratic Knapsack Problems. Mathematical Programming Study 2, 132–149 (1980)
3. Pisinger, D.: Quadratic Knapsack Problem – A Survey. Discrete Applied Mathematics 155, 623–648 (2007)
4. Singh, A., Baghel, A.S.: A Hybrid Evolutionary Approach to the Quadratic Knapsack Problem. In: Proceedings of the Third Indian International Conference on Artificial Intelligence, pp. 200–208 (2007)
5. Xie, X.-F., Liu, J.: A Mini Swarm for the Quadratic Knapsack Problem. In: Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS), pp. 190–197 (2007)
6. Xie, X.-F., Liu, J.: A Compact Multi-Agent System Based on Autonomy Oriented Computing. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 38–44 (2005)
7. Karaboga, D.: An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey (2005)
8. Karaboga, D., Basturk, B.: On the Performance of Artificial Bee Colony (ABC) Algorithm. Applied Soft Computing 8, 687–697 (2008)
9. Singh, A.: An Artificial Bee Colony Algorithm for the Leaf-Constrained Minimum Spanning Tree Problem. Applied Soft Computing 9, 625–631 (2009)
10. Billionet, A., Soutif, E.: An Exact Method Based on Lagrangian Decomposition for the 0-1 Quadratic Knapsack Problem. European Journal of Operational Research 157, 567–575 (2004)
11. Billionet, A., Soutif, E.: Using a Mixed Integer Programming Tool for Solving the 0–1 Quadratic Knapsack Problem. INFORMS Journal on Computing 16, 188–197 (2004)

# Universal Learning Machines

Włodzisław Duch and Tomasz Maszczyk

Department of Informatics, Nicolaus Copernicus University, Toruń, Poland
Google: W Duch
tmaszczyk@is.umk.pl

**Abstract.** All existing learning methods have particular bias that makes them suitable for specific kind of problems. Universal Learning Machine (ULM) should find the simplest data model for arbitrary data distributions. Several ways to create ULMs are outlined, and an algorithm based on creation of new global and local features combined with meta-learning is introduced. This algorithm is able to find simple solutions that sophisticated algorithms ignore, learn complex Boolean functions, complicated probability distributions, as well as the problems requiring multiresolution decision borders.

## 1 Introduction

Despite great progress in development of numerous new algorithms computational intelligence (CI) systems are still far behind natural biological systems in solving complex problems that face organisms, learning to optimize chances of survival, recognizing important structures in perceptual information, developing communication and coordination with other organisms. CI algorithms are quite sophisticated, but the key to general intelligence may lie in specific information filters that make learning possible, and chunking mechanisms that combine their results into higher-level mental representations. Filters discover phonemes, syllables, words in the auditory stream (with even more complex hierarchy in the visual stream), while chunking links sequences of lower level patterns into higher-level patterns, discovering associations and motifs.

On a more technical level this means that more attention should be paid to generation of features, exploiting various ways to use input information. Systematic selection and construction of new features should be followed by simple learning models that are quite accurate once an appropriate representation of the problem is found. For highly non-linear decision borders local features based on kernels localized at data samples that are close to decision borders are useful. There is some similarity to the kernel-based learning [1] that implicitly projects data into high-dimensional spaces, but here mixing different kernels and using different types of features gives much more flexibility. Cortical minicolumns extract a large number of interesting features from signals. This inspiration led to almost Random Projection Machine (aRPM) [2] algorithm where random projections generate new candidate features, and only those that contain clusters of pure data samples (in the $k$-separability sense [3]) are selected. The algorithm is fast and works quite well for complex Boole'an functions where almost all other algorithms fail. However, for some data distributions direct use of linear projections is not the best idea. For example, visual object recognition should distinguish complex shapes

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 206–215, 2009.

defined in two- or three-dimensional space. Sophisticated approaches may miss simple and comprehensible solutions.

In this paper Universal Learning Machines (ULM) are introduced, based on systematic generation and selection of features of growing complexity, that help to learn simple data models in all kinds of situations. This is the next step towards meta-learning systems [4,5], that shifts emphasis on construction of the feature space, rather than construction of sophisticated learning models. In the following section short description of meta-learning idea is given, followed by systematic description of feature constructors and some details about current implementation of the ULM approach. Illustrative results are in section 4 and a brief discussion concludes this paper.

## 2 Meta-Learning and Feature Construction

According to the ”no free lunch” theorem [6] no single system may reach the best results for all possible distributions of data. Decision trees and rule-based systems are the best for data that have simple logical structure, require sharp decision borders [7,8] but fail already on problems where linear discrimination provides accurate solution. SVM in kernelized form works well when complex topology is required but may miss simple solutions that rule-based systems find, fails when sharp decision borders are needed and fails on complex Boolean problems [9]. Each system has a particular bias that makes it suitable for particular class of problems. Discovering this bias and finding an appropriate model is not easy, and is usually done by tedious experimentations with combinations of pre-processing, filtering and selection, clusterization, classification or regression and post-processing techniques, combined with meta-learning procedures based on stacking, boosting, committees and other techniques. A number of efforts directed at automatic search for good data models are worth noting.

First approach is to search for general characteristics of data and compare it to the characteristics of reference data for which ranking of results of different system has been established [10]. However, a small change in data distribution may have a strong influence on the type of decision borders needed. The second approach tries to put various data transformations into a framework that systematically increases their complexity and do a meta-search in the space of all such models. The framework for similarity-based methods is quite suitable here, covering most classification methods, including feedforward neural networks and some kernel methods [4,5]. The granularity of the transformations use to automatically build good models is finer when heterogeneous systems are admitted, using optimized distance (kernel) functions of different types and optimized neural transfer functions in constructive network methods [11,12]. Other systems may be added to the pool of models, including heterogeneous decision trees and their forests [13]. This approach may be converted into general theory of transformation based learning, building models based on composition of transformations [14], and combining them by using committees of locally-competent models [15].

With proper control of search and complexity of generated models [16,17] such transformation-based approach offers an interesting approach that may overcome the limits of the ”no free-lunch” theorem. However, success of such meta-search for best composition of transformations relies on the availability of transformations that extract useful features and handle various specific problems, such as image analysis, multimedia streams, signal decomposition, text analysis, biosequences and many other problems. A Universal Learning Machine (ULM) is based on a very general algorithm:

- Construct new features (generate transformations).
- Test if they help to learn (generate results and select).

Learning may of course be done with one of the sophisticated schemes, but here the focus will be on construction of features. They may be ranked by some simple filter criterion [18], or be used to incrementally expand feature space using simple learning models. Analysis of all feature constructors for specific applications is beyond the scope of this paper, but it is fruitful to analyze methods that may be applied to construct new features from typical data.

## 3 Feature Construction

A set of raw initial features is provided as a part of problem description; out of these some may be directly useful and should be selected and tested. More features may be created by various transformations and pattern filters to extract useful information. A hierarchy of features with different complexity should be established, depending on the type of decision regions they provide and the complexity of search processes needed to discover them.

Binary features are the simplest and are a special case of projection on a line. Projections may be either unrestricted (all data are used for the projection), giving global features, or restricted, providing local features. Raw binary features $\{b_i\}$ should be regarded as global, resulting from unrestricted projections on a line. They are created from other features by dividing nominal features into two subsets, or creating subintervals of real features $\{x_i\}$ using decision trees or Quality of Projected Clusters [19] criteria.

Local binary features may be obtained by imposing various restrictions on projections. For binary features this will lead to complexes $b_1 \wedge b_2 ... \wedge b_k$ that help to distinguish interesting regions in feature space. For real-valued features restrictions based on intervals create hyperboxes $\prod_i [r_i^-, r_i^+]$. Restricted binary features are created by decision trees, for example if there is top level path $z = (x_1 < t_1) \wedge (x_2 \geq t_2)$, then $z$ is a binary (logical) feature made from projection on $x_2$ restricted by $x_1 < t_1$ (or vice versa). Such features are especially useful for problems with inherent logical structure [7,8]. Logical features may be smoothed and changed into real, strongly non-linear feature using sigmoidal functions $z = \sigma(t_1 - x_1)\sigma(x_2 - t_2)$. Other ways to restrict subspace used for projection may be considered, for example taking only vectors that are not far from $x_1$ line and binarizing the projection $z = \sigma(x_1 - t_1)\sigma(d - ||\boldsymbol{x}||_{-1})$, where $||\boldsymbol{x}||_{-1}$ norm excludes feature $x_1$. More general version of binary features may use different restrictions for each values $b = 0$ and $b = 1$. Similar considerations may be done for nominal features.

The real-valued raw features are sometimes directly relevant to the learning task. Enhancement of local contrast is very important in natural perception. Features transformed by a steep sigmoidal function $\sigma(\beta x_i - t_i)$ are frequently more useful, closer to binary. Slopes $\beta x_i$ and thresholds $t_i$ may be individually optimized using mutual information or other relevance measures independently for each feature. Single features may also show interesting patterns of $p(X|C)$ distributions, for example $k$ relatively large groups of values, each corresponding to the same type of objects. Such $k$-separable solutions are very useful for learning complex Boolean functions. For example, $n$-bit parity functions are $n+1$-separable if a sum of all feature values is used as a new feature. A single cluster of pure cases is worth using as a new feature. Such features are generated by applying localized window-type functions to original features [19], for example $z_i = \sigma(x_i - a_i) - \sigma(x_i - b_i), a > b$.

Thus original real features may be pre-processed in several ways to increase their usefulness. The same is true for restricted, or partially localized real features. Instead of accepting raw features additional conditions restricting the subspace from which projections are made are added, $z_{ij} = x_i$ for $|x_j| < t_j$. Linearization of 2-dimensional distributions is possible using step-functions. If the minimal value of $x_i$ is $x_{\min}$ and $z_{ij} = x_i \Theta(t_j - x_j) + (x_{\min} - 1)\Theta(x_j - t_j)$ all projections for $x_j \geq t_j$ are moved to $z_{ij} = x_{\min} - 1$. Consider for example fuzzy XOR relations between $(x_i, x_j)$; such transformation will convert it to a 4-separable problem that is easily solved by most machine learning systems.

More complex line patterns and higher-dimensional patterns may be considered. First, linear combinations of features provide frequently better features than raw features. The simplest and computationally cheapest way to generate them is to start from calculation of the class centers $\boldsymbol{m}_i$, and use normalized projection directions $\boldsymbol{w}_{ij} = (\boldsymbol{m}_i - \boldsymbol{m}_j)/||\boldsymbol{m}_i - \boldsymbol{m}_j||$. If conditional probabilities $p(\boldsymbol{x}|C)$ have approximately Gaussian distributions $z(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{w} \cdot \boldsymbol{x}$ captures all information and other features will be spurious. This is indeed the case in a number of benchmark data, making them trivial. Drawing $p(C|z)$ shows areas of significant overlaps and allows for separation of border vectors that fall near the threshold $p(C_i|z) = p(C_j|z)$. Using only those vectors more projection directions may be generated; first class-dependent clusterization of these vectors is done and then search for pairs of cluster centers from different classes that are close to each other.

In recent years kernel-based learning methods dominate in pattern recognition [1]. Kernels are used to measure similarity to reference vectors providing new features $z_i = K(\boldsymbol{x}^{(i)}, \boldsymbol{x})$. Gaussian kernels are most popular, creating features that provide localized receptive fields, measuring how far vector $\boldsymbol{x}$ is from the reference support vector $\boldsymbol{x}^{(i)}$. Suppose that distribution $p(\boldsymbol{x}) = p(x_1, x_2)$ has been created as a sum of two partially overlapping Gaussian distributions; than transforming this distribution to $(z_1, z_2)$, with kernels at the centers of clusters $z = z_1 + z_2$ variable will be almost constant along $p(\boldsymbol{x})$ =const, making the non-linear decision border almost flat after transformation. Quadratic optimization procedures in SVM may be replaced by any large-margin linear discrimination techniques, the extraction of useful information by kernels is primary reason for success.

Explicit generation of features based on different similarity measures [20] removes one SVM bottleneck, allowing for optimization of resolution in different areas of the feature space: strong non-linearities are provided where they are needed (small range of localized kernels), and using smooth functions (large range) when this is sufficient.

This technique may be called **adaptive regularization**, in contrast to simple regularization based on minimization of the norm of the weight vector $||\boldsymbol{w}||$ used in SVM or neural networks [1]. Although simple regularization enforces smooth decision borders decreasing model complexity it is not able to find the simplest solutions and may easily miss the fact that a single binary feature contains all information. Generation of kernel features should therefore proceed from most general, providing almost hyperplanar decision borders, with centers placed far from decision borders (identified looking at the $z = \boldsymbol{w} \cdot \boldsymbol{x}$ distribution for $\boldsymbol{w} = (\boldsymbol{m}_1 - \boldsymbol{m}_2)/||\boldsymbol{m}_1 - \boldsymbol{m}_2||$ direction), to more specific, highly non-linear, with non-zero contributions only close to decision border. More sophisticated features may also be useful: based on Mahalonobis distance calculated for clusters of vectors located near decision borders (an inexpensive method for rotation of density functions with $d$ parameters has been introduced in [11]), or with flat local fronts using cosine distance.

Summarizing, the (incomplete) list of feature constructors includes:

- B1: Binary – unrestricted projections; provides 4 regions $p(C|b), C = 1, 2; b = 0, 1$, for MAP classifiers, complexity $O(1)N_b$.
- B2: Binary – restricted by other binary features; complexes $b = b_1 \wedge b_2 ... \wedge b_k$; also 4 regions $p(C|b)$, full complexity $O(2^k N_b)$, but only $O(N_b)$ with greedy algorithms.
- B3: Binary – restricted by distance; $b = 0 \wedge r_1 \in [r_1^-, r_1^+] ... \wedge r_k \in [r_k^-, r_k^+]$; search is done separately for each $b$ value.
- N1: Nominal – like binary, with two or more subsets.
- R1: Line – original feature $x_i$, unrestricted projection $x_i(\boldsymbol{x})$; thresholds or intervals may change it into B1; contrast enhancements $\sigma(x_i)$, search for 1D patterns.
- R2: Line – like R1 but restricted by other features, $x_i = x_i(\boldsymbol{x})$ only for $|x_j| < t_j$.
- R3: Line – like R2 but restricted by distance, , $x_i = x_i(\boldsymbol{x})$ only for $\sum_{j \neq i} x_j^2 < t$.
- R4: Line – linear combination $z = \boldsymbol{w} \cdot \boldsymbol{x}$ optimized by unrestricted, projection pursuit (PCA, ICA, QPC ...); otherwise treated like R1.
- P: Prototype-based localized features $q(\boldsymbol{x}) = \exp(-||\boldsymbol{x} - \boldsymbol{r}||)$, weighted distance functions or specialized kernels.
- M: Motifs, based on correlations between elements (sequences, Hebbian correlations).
- T: Non-linear transformations: radial, separable, universal, rational, and other types of transfer functions [11] $f(\boldsymbol{x}) = \Phi(\boldsymbol{x}; \boldsymbol{w})$ for feature construction.

Prototype-based local features are created using support vectors near the decision border. First, projections lines $z_i = \boldsymbol{w}_i \cdot \boldsymbol{x}$ are generated using class means, and those vectors that are projected in the interval $z_i \in [\theta_i - r, \theta_i + r]$ selected, where $r = 0.05|\max z_i - minz_i|$, and the threshold $p(C|z_i = \theta_i) = p(C'|z_i)$. More line projections may be generated by grouping vectors into several clusters in each class and using restricted line projections for their means. Distances between selected vectors are evaluated (Euclidean distances are used) and only those with the fraction $[0.5 + \epsilon, 1 - \epsilon]$ of neighbors from the same class (with $\epsilon = 0.1$) in the $\sigma$ radius are left. They are close to the decision borders, but selecting those with largest $\sigma$ and clusterizing them to form features $q_i(\boldsymbol{x}) = \exp(-|\boldsymbol{x} - \boldsymbol{p}_i|/\sigma)$ will provide relatively smooth local features. Those vectors that are close to the border but have very small $q_i(\boldsymbol{x})$ are used again to create features with smaller $\sigma$, localized more strongly, until most vectors are covered by local features. In this way multi-resolution kernel projection is realized with few support vectors. Complexity of this process is dominated by distance calculation, but this is done only on a small subset of all vectors, all other steps are linear in the number of vectors.

Second-order features may be build by learning machines using features that have been constructed and selected for learning. Ranking of local features should be done with care as they may not be globally important, but will be needed for local representation of information only. ULM may be presented as a constructive network, with new nodes representing transformations and procedures to extract useful features, and additional layers analyzing the image of data in the feature space created in this way. Here feature generators and classifiers are separated. ULM starts from original features, testing consecutively binary, nominal and real features, then testing the results of adding restricted features, and constructing new features using projections and prototypes. This process may be viewed as a search in the space of various feature spaces, analogous to the search in the space of similarity-based models [4,5].

## 4 Illustrative Results

The ULM approach is illustrated using a few benchmark datasets (taken from the UCI repository [21]) that create various problems to the state-of-the-art classification systems. Three quite different systems have been used in comparison: SVM with linear and Gaussian kernels (with proper choice of P-features giving identical results), Naive Bayes (NB) classifier, and Separability-Split Value (SSV) decision tree [22], all implemented in the Ghostminer software [23]. 10-fold stratified crossvalidation tests have been done in all cases.

Reference results obtained with original features are in Table 1. In parenthesis the number of generated features is noted. Line projections R1 were generated in two ways, from calculation of the class centers or cluster centers. For Hypothyroid, Wisconsin and Australian class centers worked better, for the remaining of datasets cluster centers were used. The simplest binary features B1 extracted from decision trees are presented in Table 2.

The *Australian credit* problem (15 features, 690 samples) has been used in the Statlog project [24] that compared 22 classifiers. The best result 86.9±, has been obtained with Cal5 tree (6 leaves on average), a rather fortuitous result as this tree has never been among top three for the remaining 21 datasets used in the Statlog study, the variance in crossvalidation test is about 4%, and the results have not been properly averaged over many runs. SVM with Gaussian kernel, optimized C=0.01 and $\sigma = 0.01$, using 460 support vectors, reaches 86.2±4.1%. Starting from ranking of single features it is easy to notice that A9 (our guess is that it codes "has bank account") is by far the most important feature (ex. Pearson correlation coefficient is 0.72, while the next best feature has 0.46). Using A9 binary feature SVM results with linear kernel is 85.5±3.5% with 182 SVs used and SSV tree and NB classifier give the same accuracy. For SVM and SSV this is only slightly higher than the reference value, but for NB it is a significant (over 5%) improvement. SSV and NB solutions are equivalent to the maximum a posteriori (MAP) rule: IF A9=T then class $+$, A9=F then class $-$.

Adding more features from the original set degrades this result, also adding line projections decreases accuracy, but adding to the binary feature one prototype-based feature near the decision border improved all results in statistically significant way to 86.8%, with only 3 leaves for SSV. While adding other features did not improved the result of SVM and NB, for SSV combination of the A9 with one R1-feature and one P-feature gave even better result, 87.0±2.50 with just 4 leaves. For this dataset ULM finds not only the simplest solution, but also using additional P-feature finds the most accurate solution known so far.

The *Ljubliana cancer* data contains 286 cases, of which 201 are no-recurrence-events (base rate 70.3%) and 85 are recurrence-events (29.7%). There are 9 attributes, with 2-13 different values each. SVM with Gaussian kernel gives 73.8±4.3%, linear SVM 71.0±4.7%, MLP network reaches similar accuracy to Gaussian SVM. The most important feature is Degree-malignant$\in \{1, 2, 3\}$, that for the value 3 has more than half recurrence events. Restricting this nominal feature (equivalent to selection of one path from two-level decision tree) using the number of involved nodes (ranging from 1 to 9) creates a binary feature: $B1 =$ Degree-malignant = 3 $\wedge$ involved nodes$> 1$. The accuracy of the MAP rule based on this single feature, as well as the 3 classifiers in our study is 76.3±6.5%. Such simple and comprehensible solutions are easily missed if complex models (that in this case are less accurate) are used.

*Appendicitis* is another small dataset [7], containing only 106 cases, 88 cases with acute appendicitis and 18 cases with other problems, with 8 features (results of medical tests). Adding a single binary restricted feature B2 derived from SSV decision tree (see Table 2) increased the accuracy of all methods by 4-5%, to 91.5%, at the same time reducing complexity of classifiers. Other types of features did not improve this result.

The *Cleveland heart disease* dataset contains 303 cases, 164 corresponding to healthy (54.1%) and the rest to heart problems of various severity, described by 6 continuous and 4 nominal features. Adding new features did not lead to NB improvement in any significant way, but SVM benefited slightly from adding 3 line projections (achieving 83.5±3.6, with 98 SVs), and using 3 prototype-based features (84.2±6.1).

The *Pima Indian diabetes* dataset has only 8 attributes, 768 instances, 500 (65.1%) healthy and 268 (34.9%) diabetes cases. This dataset was used in the Statlog project [24], with the best 10-fold crossvalidation accuracy around 77.7% obtained by logistic discriminant analysis. Small improvement (1.3%) is gained in NB by adding binarized real feature (Table 2), and also by adding 3 line projections. Slight SVM improvement is noticed only with 3 line projections and 4 prototype-based features, giving 78.5±3.6%.

The *Wisconsin cancer* dataset contains 699 cases, with 458 benign (65.5%) and 241 (34.5%) malignant cases, described by 9 attributes with integer value in the range 1-10. Binarization of F2 feature improves the result in insignificantly way, but using a single line projection improves SSV results by 2.2% and NB results by 1.2%, while SVM is improved slightly only after 4 P-features are added, reaching 97.2±1.9% with 45 SVs.

The *Hypothyroid* dataset has 3772 cases for training and 3428 cases for testing, but to use consistent methodology the data has been merged and 10-fold stratified crossvalidation used. There are 22 attributes (15 binary, 6 continuous), and 3 classes: primary hypothyroid, compensated hypothyroid and normal (no hypothyroid), with unbalanced class distribution in the merged data, 166, 368 and 6666 vectors in the normal class.

Results of SVM and especially NB are quite poor (Table 1), however adding one binary restricted feature greatly improves them, from 94.1 to 99.5±0.4% with 80 SVs for SVM, and from 41.3 to 98.1±0.8% for NB. Adding more features does not improve these results further.

**Table 1.** Reference results with original features; ULM significantly improved results are described in the text

| Dataset | Classifier | | |
|---|---|---|---|
| | SVM (#SV) | SSV (#Leafs) | NB |
| Australian | 84.9±5.6 (203) | 84.9±3.9 (4) | 80.3±3.8 |
| Ljubliana cancer | 72.0±5.1 (168) | 74.7±5.0 (3) | 72.2±7.4 |
| Appendicitis | 87.8±8.7 (31) | 88.0±7.4 (4) | 86.7±6.6 |
| Heart | 82.1±6.7 (101) | 76.8±9.6 (6) | 84.2±6.1 |
| Diabetes | 77.0±4.9 (361) | 73.6±3.4 (4) | 75.3±4.7 |
| Wisconsin | 96.6±1.6 (46) | 95.2±1.5 (8) | 96.0±1.5 |
| Hypothyroid | 94.1±0.6 (918) | 99.7±0.5 (12) | 41.3±8.3 |

**Table 2.** B1 features extracted from the decision tree

| Dataset | B1 features | |
|---|---|---|
| Australian | $F8 < 0.5$ | $F8 >= 0.5$ & $F9 >= 0.5$ |
| Ljubliana cancer | Degree-malignant = 3 | Degree-malignant = 3 $\wedge$ involved nodes$>$ 1 |
| Appendicitis | $F7 >= 7520.5$ | $F7 < 7520.5$ & $F4 < 12$ |
| Heart | $F13 < 4.5$ & $F12 < 0.5$ | $F13 >= 4.5$ & $F3 >= 3.5$ |
| Diabetes | $F2 < 123.5$ | $F2 >= 143.5$ |
| Wisconsin | $F2 < 2.5$ | $F2 >= 4.5$ |
| Hypothyroid | $F17 < 0.00605$ | $F17 >= 0.00605$ & $F21 < 0.06472$ |

In case of the Wisconsin breast cancer data linear combination of features is clearly better than using individual features separately in decision trees or in Naive Bayes; in case of the hypothyroid data it is quite the opposite.

## 5 Discussion

Universal Learning Machines should be based on meta-learning schemes: searching for the best models in the space of selected data transformations. Meta-learning in the framework of similarity-based models [4,5] has been previously used with considerable success. Here a search in the space of various feature transformations is explored. Of course one may combine the two meta-learning approaches and in addition expand the space of all admissible transformations. General framework for such meta-learning schemes has recently been proposed in [16,17]. However, even the simplest scheme to construct-select features is worth exploring, generating new features that, followed by forward selection or even a simple filter method to rank all features, may dramatically simplify and improve results.

While a lot of efforts have been devoted to feature selection procedures [25] much less effort has been spent on feature construction [26]. Yet functioning of biological neural networks gives numerous examples of transformations realized by neural columns that extract non-linear features from incoming signals, reducing noise in data and learning to estimate similarity of responses. Neural columns learn to react not only to intensity, but also to specific structures in the incoming stimuli, solving complex perceptual problems. As a step towards universal learning machines various ways of extracting information from input data have been systematically described, from binary features and their enhancements, through combinations of features derived from simple decision trees, combinations of features estimated from clusters, localized clusters after projections, to kernel-based features at different level of granularity, allowing for adaptive regularization. This of course does not exhaust all possibilities for construction of informative features. Various projection pursuit networks that reduce dimensionality, including PCA, ICA and other techniques [27], create interesting combination of original raw features according to such criteria as maximum variance or independence.

Feature constructors described here go beyond linear combinations. Systematic explorations of features of growing complexity allows for discovery of simple models that more sophisticated learning systems will miss. Some benchmark problems have been found rather trivial, and have been solved with a single binary feature, one constrained nominal feature, or one new feature constructed as a projection on a line connecting means of two classes. Kernel-based features offer an attractive alternative to current kernel-based SVM approaches, offering multiresolution and adaptive regularization possibilities. Analysis of images, multimedia streams or biosequences will require even more sophisticated ways of constructing features based on simpler features. Although larger datasets should be analyzed it is quite clear that constructing new features in this way opens new possibilities to create simple, comprehensible and accurate data models.

## References

1. Schölkopf, B., Smola, A.: Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2001)
2. Duch, W., Maszczyk, T.: Almost random projection machine. In: Alippi, C., et al. (eds.) ICANN 2009, Part I. LNCS, vol. 5768, pp. 789–798. Springer, Heidelberg (2009)
3. Duch, W.: $k$-separability. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4131, pp. 188–197. Springer, Heidelberg (2006)
4. Duch, W., Grudziński, K.: Meta-learning: searching in the model space. In: Proceedings of the International Conference on Neural Information Processing, Shanghai, pp. 235–240 (2001)
5. 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. Springer-Physica Verlag, New York (2002)
6. Duda, R.O., Hart, P.E., Stork, D.: Patter Classification. J. Wiley & Sons, New York (2001)
7. Duch, W., Adamczak, R., Grąbczewski, K.: A new methodology of extraction, optimization and application of crisp and fuzzy logical rules. IEEE Transactions on Neural Networks 12, 277–306 (2001)

8. Duch, W., Setiono, R., Zurada, J.: Computational intelligence methods for understanding of data. Proceedings of the IEEE 92, 771–805 (2004)
9. Grochowski, M., Duch, W.: Learning highly non-separable Boolean functions using Constructive Feedforward Neural Network. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4668, pp. 180–189. Springer, Heidelberg (2007)
10. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning: Applications to Data Mining. In: Cognitive Technologies. Springer, Heidelberg (2009)
11. Duch, W., Jankowski, N.: Survey of neural transfer functions. Neural Computing Surveys 2, 163–213 (1999)
12. Duch, W., Jankowski, N.: Taxonomy of neural transfer functions. In: International Joint Conference on Neural Networks, Como, Italy, vol. III, pp. 477–484. IEEE Press, Los Alamitos (2000)
13. Duch, W.: Grąbczewski, K.: Heterogeneous adaptive systems. In: IEEE World Congress on Computational Intelligence, pp. 524–529. IEEE Press, Honolulu (2002)
14. Duch, W.: Towards comprehensive foundations of computational intelligence. In: Duch, W., Mandziuk, J. (eds.) Challenges for Computational Intelligence, vol. 63, pp. 261–316. Springer, Heidelberg (2007)
15. Duch, W., Itert, L.: Committees of undemocratic competent models. In: Rutkowski, L., Kacprzyk, J. (eds.) Proc. of Int. Conf. on Artificial Neural Networks (ICANN), Istanbul, pp. 33–36 (2003)
16. Grabczewski, K., Jankowski, N.: Versatile and efficient meta-learning architecture: Knowledge representation and management in computational intelligence. In: IEEE Symposium on Computational Intelligence in Data Mining, pp. 51–58. IEEE Press, Los Alamitos (2007)
17. Grabczewski, K., Jankowski, N.: Meta-learning with machine generators and complexity controlled exploration. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 545–555. Springer, Heidelberg (2008)
18. Duch, W.: Filter methods. In: Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds.) Feature extraction, foundations and applications, pp. 89–118. Springer-Physica Verlag, New York (2006)
19. Grochowski, M., Duch, W.: Projection Pursuit Constructive Neural Networks Based on Quality of Projected Clusters. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) ICANN 2008, Part II. LNCS, vol. 5164, pp. 754–762. Springer, Heidelberg (2008)
20. 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)
21. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
22. Grąbczewski, K., Duch, W.: The separability of split value criterion. In: Proceedings of the 5th Conf. on Neural Networks and Soft Computing, Zakopane, Poland, Polish Neural Network Society, pp. 201–208 (2000)
23. Duch, W., Jankowski, N., Grąbczewski, K., Naud, A., Adamczak, R.: Ghostminer data mining software. Technical report, Department of Informatics, Nicolaus Copernicus University (2000-2008), http://www.fqspl.com.pl/ghostminer/
24. Michie, D., Spiegelhalter, D.J., Taylor, C.C.: Machine learning, neural and statistical classification. Elis Horwood, London (1994)
25. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: Feature extraction, foundations and applications. Springer-Physica Verlag, Heidelberg (2006)
26. Liu, H., Motoda, H.: Feature extraction, construction and selection: a data mining perspective. In: Liu, H., Motoda, H. (eds.) SECS, vol. 453. Kluwer Academic, Boston (1998) (Includes bibliographical references and index)
27. Pękalska, E., Duin, R.: The dissimilarity representation for pattern recognition: foundations and applications. World Scientific, Singapore (2005)

# Swarm Diversity Based Text Summarization

Mohammed Salem Binwahlan[1], Naomie Salim[2], and Ladda Suanmali[3]

[1,2] Faculty of Computer Science and Information Systems,
Universiti Teknologi Malaysia, 81310, Skudai, Johor, Malaysia, +60177582859
moham2007med@yahoo.com, naomie@utm.my
[3] Faculty of Science and Technology,
Suan Dusit Rajabhat University 295 Ratchasrima Rd, Dusit, Bangkok, Thailand 10300
ladda_sua@dusit.ac.th

**Abstract.** Automatic text summarization systems aim to make their created summaries closer to human summaries. The summary creation under the condition of the redundancy and the summary length limitation is a challenge problem. The automatic text summarization system which is built based on exploiting of the advantages of different techniques in form of an integrated model could produce a good summary for the original document. In this paper, we introduced an integrated model for automatic text summarization problem; we tried to exploit different techniques advantages in building of our model like advantage of diversity based method which can filter the similar sentences and select the most diverse ones and advantage of the differentiation between the most important features and less important using swarm based method. The experimental results showed that our model got the best performance over all methods used in this study.

**Keywords:** Binary tree, Diversity, MMI, Summarization, Swarm, Summary.

## 1 Introduction

Dramatic increasing of the information amount on the web made the finding of the interesting information more difficult. The information search tools or search engines became ineffective in facing such problem in the user's point of view. Due to this problem, it was imposed those information search tools to employ useful helping factor to get ride of such difficulty. Up to now the automatic text summarization is the main useful helping factor for those information search tools which produces a short summary of  the content of each explored web page. Automatic text summarization is a technique concerning the creation of compressed form for single document or multi-documents. The aim of automatic text summarization techniques is to find the most important text units and present them as summary of the original document. Each technique differs from another in the way of discovering such text units. The automatic text summarization system which is built based on exploiting of the advantages of different techniques in form of an integrated model could produce a good summary for the original document. Many techniques have been proposed for automatic text summarization problem based on different methodologies, [1; 2; 3] used the shallow

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 216–225, 2009.

features to score the text units and selecting the highest score text units as summary, [4; 5; 6; 7; 8; 9; 10] could add advantage to the mechanism of the text unit scoring which is the exploitation of the data to create a criteria or weights to be used in the scoring coefficient through the applying of the machine learning for automatic text summarization problem. The discourse structure based techniques [11; 12; 13] employed the discourse structure for the sentence scoring.

All techniques mentioned above did not pay attention to the problem of the redundancy which causes the low quality of the created summary. The method which was built for dealing with the problem of redundancy is MMR [14], many methods made use of MMR either directly or after modifying it, [15; 16; 17; 18; 19; 20; 21].

The improvement of the summary quality remains the key research problem and needs much work like incorporate more than one good technique. Aretoulaki [22] proposed a hybrid system, the system was built based on four modules, where each module tries to look for specific features and information in the input text, then the outputs of those modules are passed to Artificial Neural Network (ANN) to score the text units as important and unimportant based on those outputs of the four modules. Alemany and Fort [23] presented a summarizer based on lexical chains, in which, the cohesive properties of the text were combined with coherence relations to produce good summaries. a different hybrid model was introduced by Cunha et al. [24], which combines mainly three systems, each system produces its own extract, then an algorithm creates the final summary by selecting the highest score sentences from the three extracts after scoring of those extract sentences. The summary creation under the condition of the redundancy and the summary length limitation is a challenge problem. Therefore in this paper, we introduce an integrated model for automatic text summarization problem; we try to exploit different techniques advantages in building of our model like advantage of diversity based method [25] which can filter the similar sentences and select the most diverse ones and advantage of the differentiation between the most important features and low important using swarm based method [26].

The rest of this paper is organized as follows: Section 2 presents MMI diversity based text summarization method, swarm based text summarization method and introduces integrating of the MMI diversity based text summarization and swarm based text summarization. Section 3 discusses experimental design. Section 4 and 5 presents the method and experimental results. Section 6 presents discussion. Section 7 gives conclusion and future work.

## 2 MMI Diversity Based Text Summarization

MMI (maximal marginal importance) [25], it is a text summarization diversity based method for summary generation. It depends on the extraction of the highest important sentences from the original text. The text features used in this method are: sentence centrality, title-help sentence (THS), title-help sentence relevance sentence (THSRS), word sentence score (WSS), key word feature and the similarity to first sentence.

To summarize a document using this method, it is required first to cluster the document sentences into clusters (using k-means clustering algorithm) where each cluster contains the most similar sentences. The clusters number is determined automatically by the summary length (number of sentences in the final summary). Each sentences cluster is represented as one binary tree or more. The first sentence which is presented in the binary tree is that sentence with higher number of friends (higher number of similar sentences), then the sentences which are most similar to already presented sentence are selected and presented in the same binary tree. The sentences in the binary tree are ordered based on their scores. The score of the sentence in the binary tree building process is calculated based on the importance of the sentence and the number of its friends using eq. 1.

$$\text{Score}_{BT}(s_i)=\text{impr}(s_i)+(1-(1-\text{impr}(s_i)\times\text{friendsNo}(s_i))) \tag{1}$$

Where $\text{Score}_{BT}(s_i)$ is the score of the sentence $s_i$ in the binary tree building process, $\text{impr}(s_i)$ is the importance of the sentence $s_i$ calculated using normal features eq. 2 and friendsNo($s_i$) is the number of sentences which are similar to the sentence $s_i$.

$$\text{impr}(S_i)=\text{avg}(\text{WSS}(S_i)+\text{SC}(S_i)+\text{SS\_NG}(S_i)+\text{sim\_fsd}(S_i)+\text{kwrd}(S_i)) \tag{2}$$

Where WSS: word sentence score, SC: sentence centrality, SS_NG: average of THS and THSRS features, sim_fsd: the similarity of the sentence $s_i$ with the first document sentence calculated using cosine similarity measure and kwrd($s_i$) is the key word feature.

Each level in the binary tree contains $2^{\text{ln}}$ of the higher score sentences, where ln is the level number, ln=0, 1, 2, .., n, the top level contains one sentence which is a sentence with highest score. In case, there are sentences remaining in the same cluster, a new binary tree is built for them by the same procedure.

MMI is used to select one sentence from the binary tree of each sentence cluster to be included in the final summary. In the binary tree, a level penalty is imposed on each level of sentences which is 0.01 times the level number. The purpose of the level penalty is to reduce the noisy sentences score. The summary sentence is selected from the binary tree by traversing all levels and applying MMI on each level sentences.

$$\text{MMI}(S_i)=\text{Arg}\max_{S_i\in CS\backslash SS}\left[(\text{Score}_{BT}(S_i)-\beta(S_i))-\max_{S_j\in SS}(\text{Rel}(S_i,S_j))\right] \tag{3}$$

Where:

$$\text{Rel}(s_i,s_j)=\text{avg}(\text{nfriends}(s_i,s_j)+\text{ngrams}(s_i,s_j)+\text{sim}(s_i,s_j)) \tag{4}$$

Where $\text{Rel}(s_i,s_j)$ is the relevance between the two competitive sentences, $s_i$ is the unselected sentence in the current binary tree, $s_j$ is the already selected sentence, ss is the list of already selected sentences, cs is the competitive sentences of the current binary tree, β is the penalty level and $\text{Nfriends}(s_i,s_j)$ is the shared friends (the group of sentences which are similar to both sentences $s_i$ and $s_j$), $\text{ngrams}(s_i,s_j)$ is the shared ngrams (the group of ngrams which are contained in both sentences $s_i$ and $s_j$) and $\text{sim}(s_i,s_j)$ is the similarity between those two sentences.

## 3 Swarm Based Text Summarization

The swarm based text summarization method [26] generates a summary of the original document by picking up the top n sentences which have high scores. Where n equal the predefined summary length and the sentences are scored using the same features presented in section 2.1. The score of each feature is adjusted by a weight. The features weights are generated by training of the particle swarm optimization (PSO) [27]. 100 documents were selected from Document Understanding Conference (DUC) [28] data collection, DUC 2002 and used as training and testing data. The swarm method is defined as combination of adjusted features scores as in (5)

$$\text{Score}(s)=\sum_{i=1}^{5} w_i \times \text{score\_f}_i(s) \tag{5}$$

Where Score(s) is the score of the sentence s, $w_i$ is the weighted of the feature i produced by PSO, i=1 to 5 and score_f$_i$(s) is the score of the feature i.

## 4 Methodology

The proposed model for summary creation depends on the extraction of the highest important sentences from the original text; we try to investigate the performance of integrating of two methods: MMI diversity based Text Summarization [25] and Swarm Based Text Summarization [26], Figure 1.

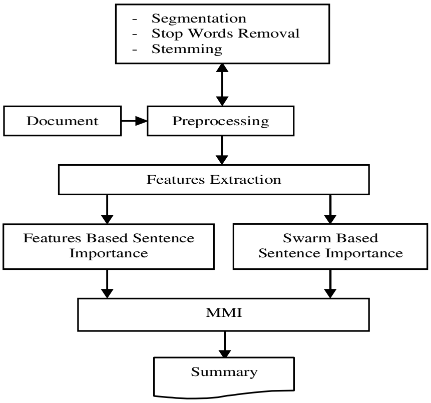
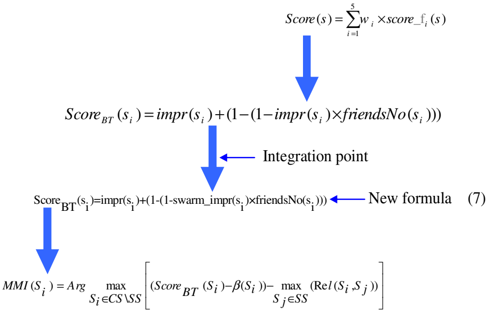


**Fig. 1.** Integrated model

In MMI method, the score of the sentence in the binary tree is calculated using:

$$\text{Score}_{BT}(s_i)=\text{impr}(s_i)+(1-(1-\text{impr}(s_i)\times\text{friendsNo}(s_i))) \tag{6}$$

In this equation the importance of the sentence appears in two positions, both two importances of the sentence are calculated using a simple combination of the text features score. We replaced the sentence importance in the second position by the sentence importance which is calculated using swarm weights based adjusted features scores. The steps of the integration process are illustrated in figure 2.



$$Score(s) = \sum_{i=1}^{5} w_i \times score\_f_i(s)$$

$$Score_{BT}(s_i) = impr(s_i) + (1 - (1 - impr(s_i) \times friendsNo(s_i)))$$

$$\mathrm{Score}_{\mathrm{BT}}(\mathrm{s}_\mathrm{i})=\mathrm{impr}(\mathrm{s}_\mathrm{i})+(1-(1-\mathrm{swarm\_impr}(\mathrm{s}_\mathrm{i})\times\mathrm{friendsNo}(\mathrm{s}_\mathrm{i}))) \qquad (7)$$

$$MMI(S_i) = Arg \max_{S_i \in CS \setminus SS} \left[ (Score_{BT}(S_i) - \beta(S_i)) - \max_{S_j \in SS} (\mathrm{Re}l(S_i, S_j)) \right]$$

**Fig. 2.** The steps of the integration process

Where in eq. 7 swarm_impr($s_i$) is the importance of the sentence $s_i$ calculated using eq. 5.

The reason of making the features are the centre of the integrating of the two methods is because the features are the cornerstone in the generation process of the text summary. The summary quality is sensitive for those features in terms of how the sentences are scored based on the used features. Therefore the exploiting of the advantages of different techniques can be a good way for evaluating the sentences.

## 5 Results and Evaluation

The DUC 2002 document sets (D061j, D062j, D063j, D064j, D065j, D066j, D067f, D068f, D069f, D070f, D071f, D072f, D073b and D077b) comprising 100 documents were used for testing [28]. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) toolkit [29] is used for evaluation, where ROUGE compares a system generated summary against a human generated summary to measure the quality. ROUGE is the main metric in the DUC text summarization evaluations. It has different variants, in our experiment, we use ROUGE-N (N=1 and 2) and ROUGE-L. In DUC 2002 document sets, each document set contains two model or human generated summaries for each document. We gave the names H1 and H2 for those two model summaries. The human summary H2 is used as benchmark to measure the quality of our proposed model summary, while the human summary H1 is used as reference summary. Beside the human with human benchmark (H2-H1) (H2 against H1); we also use another benchmark which is MS word summarizer (Msword).

We ran three experiments using the same data set, first is MMI diversity based method, second is swarm based method, and third is the integrating of the two previous methods (MMI diversity based method, swarm based method). We present the three evaluation measures ROUGE-1, ROUGE-2 and ROUGE-L with the metrics recall, precision and f-measure.

The results are shown in the tables 1, 2 and 3 using ROUGE-1, ROUGE-2 and ROUGE-L respectively. We can see in table 1 for example there is a big difference between recall and precision; therefore, we will compare all methods based on the average f-measure evaluation which is a balance between recall and precision. Figures 3,4, and 5 draw the same results.

Based on the average f-measure evaluation for ROUGE-1, table 1, the swarm based method is better than MMI diversity based method but the advantage of MMI diversity based method is that it takes into account the redundancy problem while the swarm based method does not consider that problem. The integrating of the MMI diversity and swarm based methods outperforms them individually. The benchmark Msword got less performance than the individual methods and integrated method. The same thing can be said on the performance of the methods for ROUGE-2 in table 2 and for ROUGE-L in table 3.

**Table 1.** MMI, swarm, integrated model, msword summarizer and H2-H1 comparison: average recall, average precision and average f-measure using rouge-1 at the 95%-confidence interval

| METHOD | AVG-R | AVG-P | AVG-F |
|---|---|---|---|
| Sword | 0.39306 | 0.48487 | 0.42477 |
| MMI | 0.42288 | 0.48837 | 0.4442 |
| Swarm | 0.43028 | 0.47741 | 0.44669 |
| Integrated model | **0.42678** | **0.49368** | **0.44848** |
| H2-H1 | 0.49657 | 0.49613 | 0.49605 |

**Table 2.** MMI, swarm, integrated model, msword summarizer and H2-H1 comparison: average recall, average precision and average f-measure using rouge-2 at the 95%-confidence interval

| METHOD | AVG-R | AVG-P | AVG-F |
|---|---|---|---|
| Msword | 0.16325 | 0.21066 | 0.17947 |
| MMI | 0.18213 | 0.22242 | 0.19504 |
| Swarm | 0.18828 | 0.21622 | 0.19776 |
| Integrated model | **0.18629** | **0.22817** | **0.19971** |
| H2-H1 | 0.20957 | 0.2094 | 0.20938 |

**Table 3.** MMI, swarm, integrated model, msword summarizer and H2-H1 comparison: average recall, average precision and average f-measure using rouge-L at the 95%-confidence interval

| METHOD | AVG-R | AVG-P | AVG-F |
|---|---|---|---|
| Msword | 0.36605 | 0.45272 | 0.39604 |
| MMI | 0.38865 | 0.45131 | 0.40889 |
| Swarm | 0.39674 | 0.44143 | 0.41221 |
| Integrated model | **0.39261** | **0.45692** | **0.4133** |
| H2-H1 | 0.46524 | 0.4649 | 0.46479 |

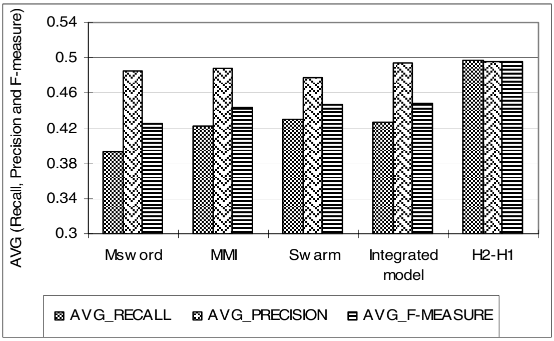


**Fig. 3.** MMI, swarm, integrated model, msword summarizer and H2-H1 comparison: average recall, average precision and average f-measure using rouge-1 at the 95%-confidence interval

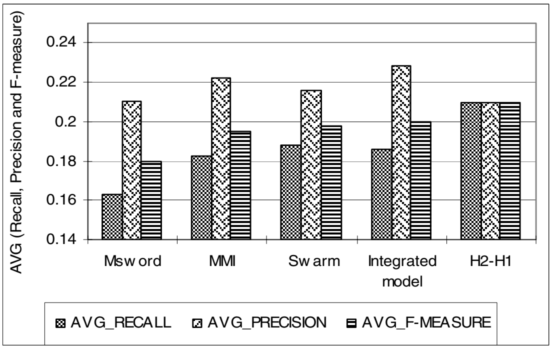


**Fig. 4.** MMI, swarm, integrated model, msword summarizer and H2-H1 comparison: average recall, average precision and average f-measure using rouge-2 at the 95%-confidence interval

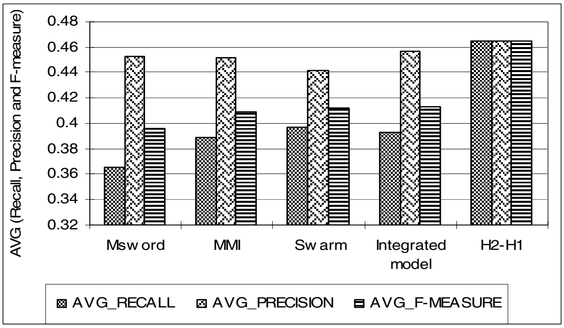


**Fig. 5.** MMI, swarm, integrated model, msword summarizer and H2-H1 comparison: average recall, average precision and average f-measure using rouge-L at the 95%-confidence interval

## 6 Discussion

In this experiment, we have tested the hypothesis of exploiting of the advantages of different techniques in form of an integrated model could produce a good summary, it was not reject. The integration point of the two methods (MMI diversity based method and swarm based method) has been made in the scoring phase of the sentences because the sentences scoring is the unique way for determining the most important ideas in the text. The swarm based method played an important role in the differentiation between the most important features and low important using the weights produced by PSO. MMI diversity based method could filter the similar sentences and select the most diverse ones. The experimental results of the proposed model testing have shown good performance when comparing with the two methods (MMI diversity based method and where swarm based method) and the benchmark methods used in this study.

## 7 Conclusion and Future Work

In this study, we introduced an integrated model for automatic text summarization problem. We built our model based on the integrating of two techniques (the first technique is diversity based and the other technique is non diversity based). The proposed model has advantage of exploiting of the abilities of different resources. The advantage of the diversity based methods is their ability to filter the most similar sentences and select the most diverse ones and advantage of the non diversity method used in this study is the differentiation between the most important features and less important. The experimental results have shown that the proposed model performs well. For future work, we plan to introduce a different hybrid model of the two techniques combined in the current model based on a different methodology for creating the summaries.

## Acknowledgments

This project is sponsored partly by the Ministry of Science, Technology and Innovation under E-Science grant 01-01-06—SF0502, Malaysia.

## References

1. Luhn, H.P.: The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development 2(92), 159–165 (1958)
2. Baxendale, P.: Machine-made index for technical literature - an experiment. IBM Journal of Research Development 2(4), 354–361 (1958)
3. Edmundson, H.P.: New methods in automatic extracting. Journal of the Association for Computing Machinery 16(2), 264–285 (1969)
4. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: Proceedings of the ACM SIGIR conference, New York, USA, July 1995, pp. 68–73 (1995)
5. Lin, C.Y., Hovy, E.: Identifying topics by position. In: Proceedings of the Fifth conference on applied natural language processing, San Francisco, CA, USA, March 1997, pp. 283–290 (1997)

6. Lin, C.Y.: Training a selection function for extraction. In: Proceedings of the Eighteenth Annual International ACM Conference on Information and Knowledge Management (CIKM), Kansas City, Kansas, November 2-6, 1999, pp. 55–62 (1999)
7. Conroy, J.M., O'leary, D.P.: Text summarization via hidden markov models. In: Proceedings of SIGIR 2001, September 9-12, 2001, pp. 406–407 (2001)
8. Osborne, M.: Using maximum entropy for sentence extraction. In: Proceedings of the ACL 2002 Workshop on Automatic Summarization, Morristown, NJ, USA, July 2002, pp. 1–8 (2002)
9. Svore, K., Vanderwende, L., Burges, C.: Enhancing single-document summarization by combining RankNet and third-party sources. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 2007, pp. 448–457. Association for Computational Linguistics, Prague (2007)
10. Fattah, M.A., Ren, F.: GA, MR, FFNN, PNN and GMM based models for automatic text summarization. Computer Speech and Language 23(1), 126–144 (2008)
11. Ono, K., Sumita, K., Miike, S.: Abstract generation based on rhetorical structure extraction. In: Proceedings of 15th International Conference on Computational Linguistics (COLING 1994), Kyoto, August 5-9, 1994, pp. 344–348 (1994)
12. Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS 1997), August 1997, pp. 10–17. ACL, Madrid (1997)
13. Marcu, D.: Improving summarization through rhetorical parsing tuning. In: Proceedings of The Sixth Workshop on Very Large Corpora, Montreal, Canada, August 1998, pp. 206–215 (1998a)
14. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: SIGIR 1998: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, August 24-28, 1998, pp. 335–336 (1998)
15. Kraaij, W., Spitters, M., Heijden, M.v.d.: Combining a mixture language model and naive bayes for multi-document summarization. In: Proceedings of Document Understanding Conference, New Orleans, LA, September 13-14, pp. 109–116 (2001)
16. Mori, T., Nozawa, M., Asada, Y.: Multi-Answer-Focused Multi-document Summarization Using a Question-Answering Engine. ACM Transactions on Asian Language Information Processing 4(3), 305–320 (2005)
17. Liu, D., Wang, Y., Liu, C., Wang, Z.: Multiple Documents Summarization Based on Genetic Algorithm. In: Wang, L., et al. (eds.) Fuzzy Systems and Knowledge Discovery, pp. 355–364. Springer, Heidelberg (2006)
18. Zajic, D.M., Dorr, B.J., Schwartz, R., Lin, J.: Sentence Compression as a Component of a Multi-Document Summarization System. In: Proceedings of the 2006 Document Understanding Workshop, June 8-9 (2006)
19. Filippova, K., Mieskes, M., Nastase, V., Ponzetto, S.P., Strube, M.: Cascaded Filtering for Topic-Driven Multi-Document Summarization. In: Proceedings of the Document Understanding Conference, Rochester, N.Y., April 26-27, pp. 30–35 (2007)
20. Ye, S., Qiu, L., Chua, T., Kan, M.: NUS at DUC 2005: Understanding documents via concept links. In: Proceedings of Document Understanding Conference, Vancouver, Canada, October 9-10 (2005)
21. Lin, Z., Chua, T., Kan, M., Lee, W., Sun, Q.L., Ye, S.: NUS at DUC 2007: Using Evolutionary Models of Text. In: Proceedings of Document Understanding Conference, Rochester, NY, USA, April 26-27 (2007)

22. Aretoulaki, M.: Towards a Hybrid Abstract Generation System. In: Int. Conf. on New Methods in Language Processing, Manchester, pp. 220–227 (1994)
23. Alemany, A.L., Fort, M.F.: Integrating cohesion and coherence for Automatic Summarization. In: EACL 2003 Student Session, ACL, Budapest, pp. 1–8 (2003)
24. da Cunha, I., Fernández, S., Velázquez Morales, P., Vivaldi, J., SanJuan, E., Torres-Moreno, J.-M.: A new hybrid summarizer based on vector space model, statistical physics and linguistics. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 872–882. Springer, Heidelberg (2007)
25. Binwahlan, M.S., Salim, N., Suanmali, L.: MMI Diversity Based Text Summarization. IJCSS International Journal of Computer Science and Security 3(1), 23–33 (2009)
26. Binwahlan, M.S., Salim, N., Suanmali, L.: Swarm Based Text Summarization. In: International Conference on IACSIT Spring Conference, Singapore, Singapore, April 17-20, 2009, pp. 145–150 (2009)
27. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm, Systems Man, and Cybernetics. In: IEEE International Conference on Computational Cybernetics and Simulation, New York, vol. 5, pp. 4104–4108 (1997)
28. NIST, The Document Understanding Conference (DUC). (2002), `http://duc.nist.gov`
29. Lin, C.: Rouge: A package for automatic evaluation of summaries. In: Proceedings of the Workshop on Text Summarization Branches Out, 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, July 25-26, pp. 74–81 (2004b)

# A Fuzzy Bi-level Pricing Model and a PSO Based Algorithm in Supply Chains

Ya Gao[1], Guangquan Zhang[1], Jie Lu[1], and Hui-Ming Wee[2]

[1] Faculty of Engineering & Information Technology,
University of Technology, Sydney,
PO Box 123, NSW 2007, Australia
{yagao,zhangg,jielu}@it.uts.edu.au
[2] Department of Industrial and Systems Engineering,
Chung Yuan Christian University,
Chungli 32023, Taiwan, ROC
weehm@cycu.edu.tw

**Abstract.** Due to rapid technological innovation and severe competition, the upstream component price and the downstream product cost in hi-tech industries usually decline significantly with time. In building a pricing supply chain model, some coefficients are generally obtained from experiments and cannot be defined as crisp numbers. Thus, an effective fuzzy pricing supply chain model becomes crucial. This paper establishes a fuzzy bi-level pricing model for buyers and vendors in supply chains. Then, a particle swarm optimization (PSO) based algorithm is developed to solve problems defined by this model. Experiments show that this PSO-based algorithm can solve fuzzy bi-level pricing problems effectively.

**Keywords:** Two-stage supply chain, bi-level programming, hierarchical decision-making, optimization, particle swarm optimization, fuzzy set.

## 1 Introduction

Hi-tech products have the following characters: there are shorter product life cycle time, quicker responsive time, increasing need for globalization and massive customization. Moreover, the material purchase cost and the product market price are decreasing at a continuous and sustained rate. The lead-time from order to delivery is usually suppressed from 955 (delivery 95% order within 5 days) to 1002 (delivery 100% order within 2 days) [1]. In some Hi-tech industries such as computers and communication consumer's products, some component costs and product prices are declining at about 1% per week [2]. This implies that purchasing or selling one-week earlier or later will result in about 1% loss.

Many researchers like Lev and Weiss [3], Goyal [4] and Gascon [5] have studied the ordering policy in the classic economic order quantity (EOQ) model for finite and infinite horizon. Buzacott [6] and Erel [7] considered a continuous price increase due to inflation. Buzacott [6] assumed compound increasing price and setup cost with inflation in a finite horizon. Erel [7] considered a compound-increasing price EOQ

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 226–233, 2009.

model with inflation rate. Yang and Wee [8] addressed a quick response production strategy with continuous demand and price declining in a finite horizon. Khouja and Park [9] derived an optimal lot size model for a decreasing rate of purchase cost in a finite horizon. All these researches with cost/price change were based on the case of single echelon. Most of these traditional EOQ models consider only the buyer's own profit. Recently, Yang el al. [1] developed a collaborative pricing and replenishment policy which takes into consideration the perspectives of a vendor and a buyer simultaneously. However, in reality, a buyer and a vendor in a supply chain would have competitive relationship in nature. It is very difficult for them to totally share interests and prices collaboratively. They need to make decisions based on their own interests but also have to consider the decisions from the other part as it affects their own interest as well.

When defining a pricing model in a supply chain, the coefficients, such as costs from a buyer or a vendor are often required to be obtained through some experiments and/or some experts' understanding of the nature of the coefficients. It has been observed that, in most real-world situations, the possible values of these coefficients are often imprecisely or ambiguously known to the experts and cannot be described precisely. With this observation, it would be certainly more appropriate to interpret the experts' understanding of the coefficients as fuzzy numerical data which can be represented by means of fuzzy sets [10].

Bi-level programming techniques aim to solve problems where each decision entity independently optimizes its own objective but is affected by the actions of other entities. In a bi-level decision problem, a decision entity at the upper level is known as the leader, and at the lower level, the follower [11]. Bi-level problems have been studied widely [12]. The investigation of bi-level problems is strongly motivated by real world applications, and bi-level programming techniques have been applied with remarkable success in different domains such as decentralized resource planning [13], electric power market [14], logistics [15], civil engineering [16], and road network management [17][18].

To practically solve the pricing problem in a supply chain , this paper uses bi-level techniques to develop a fuzzy bi-level pricing model. This model considers a buyer as the leader and a vendor as the follower, allowing them to make decisions sequentially; fully considering the mutual influences. That is, both the buyer and the vendor aim to maximize their profits in the supply chain system but their decisions are related with each other in a hierarchical way.

This paper is organized as follows. After the introduction in Section 1, Section 2 introduces related definitions. In Section 3, we build a fuzzy bi-level pricing model for the buyers and the vendors in a supply chain. To solve the problems defined by this model, a PSO based algorithm is developed in Section 4. Section 5 employs an example to carry out the experiments. Finally, conclusions and further studies are highlighted in Section 6.

## 2 Preliminary

Definition 2.1[19] Let $\tilde{a},\tilde{b}\in F(R)$ be two fuzzy numbers, the ranking relationship between $\tilde{a}$ and $\tilde{b}$ is defined as follows:

$$\tilde{a} \le \tilde{b} \text{ if } m(\tilde{a}) < m(\tilde{b}) \text{ or } m(\tilde{a}) = m(\tilde{b}) \text{ and } \sigma(\tilde{a}) \ge \sigma(\tilde{b})$$

where the mean $m(\tilde{a})$ and the standard deviation $\sigma(\tilde{a})$ are defined as

$$m(\tilde{a}) = \frac{\int_{s(\tilde{a})} x\tilde{a}(x)dx}{\int_{s(\tilde{a})} \tilde{a}(x)dx}$$

$$\sigma(\tilde{a}) = \left( \frac{\int_{s(\tilde{a})} x^2\tilde{a}(x)dx}{\int_{s(\tilde{a})} \tilde{a}(x)dx} - (m(\tilde{a}))^2 \right)^{1/2}$$

where $s(\tilde{a}) = \{x \mid \tilde{a}(x) > 0\}$ is the support set of fuzzy number $\tilde{a}$.

For triangular fuzzy number $\tilde{a} = (l, m, n)$,

$$m(\tilde{a}) = \frac{1}{3}(l + m + n)$$

$$\sigma(\tilde{a}) = \frac{1}{18}(l^2 + m^2 + n^2 - lm - ln - mn)$$

## 3 A Fuzzy Bi-level Pricing Model

In this section, we develop a fuzzy bi-level pricing model for a buyer and a vendor in a supply chain.

In a buyer-vendor system, a buyer's net profit can be calculated from [19] as follows:

$$\max_{m \in M} NP_b(m) = \frac{P_{m0}\tilde{D}}{\ln(1-r_m)}[e^{H\ln(1-r_m)} - 1] - P_{b0}Q\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} - \frac{F_b H P_{b0} Q}{2mn}\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} - mn\tilde{C}_b \tag{1}$$

A vendor's net profit can be calculated from [19] as follows:

$$\max_{n \in N} NP_v(n) = P_{b0}Q\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} - P_{v0}mQ\frac{1-(1-r_v)^H}{1-(1-r_v)^{\frac{H}{n}}} - \frac{F_v H P_{v0}(m-1)Q}{2n}\frac{1-(1-r_v)^H}{1-(1-r_v)^{\frac{H}{n}}} - n\tilde{C}_v \tag{2}$$

In (1), a buyer controls *m*, the number of buyer's lot size deliveries per vendor's lot size; and *rm*, the weekly decline-rate of market price to the end-consumer. In (2), a vendor controls *n*, the number of orders that a vendor places for the item from a supplier in the planning horizon; *rb*, the weekly decline-rate of the buyer's purchase cost; and *rv*, the weekly decline-rate of the vendor's purchase cost. All other parameters defined in the problem are constants. The explanations of symbols used in the above two formulas are listed in Table 1.

**Table 1.** Explanations on symbols used in (1) and (2)

| | |
|---|---|
| *n* | number of orders that a vendor places for the item from a supplier in the planning horizon |
| *m* | number of buyer's lot size deliveries per vendor's lot size |
| *Q* | buyer's lot size |
| *rb* | weekly decline-rate of the buyer's purchase cost |
| *D* | weekly demand rate |
| *rv* | weekly decline-rate of the vendor's purchase cost |
| *rm* | weekly decline-rate of market price to the end-consumer |
| *H* | weekly length of the planning horizon |
| *Fv* | vendor's holding cost per dollar per week |
| *Fb* | buyer's holding cost per dollar per week |
| *Cv* | vendor's ordering cost per order |
| *Cb* | buyer's ordering cost per order |
| *Pv0* | vendor's unit purchase cost at the initial time |
| *Pb0* | buyer's unit purchase cost at the initial time |
| *Pm0* | market price to the end consumer at the initial time |
| *Pv(t)* | vendor's unit purchase cost in week t |
| *Pb(t)* | buyer's unit purchase cost in week t |
| *Pm(t)* | market price to the end consumer in week t |
| *NPv* | vendor's net profit in the planning horizon |
| *NPb* | buyer's net profit in the planning horizon |
| *NP* | joint net profit of both the vendor and the buyer in the planning horizon |

In reality, the demand rate *D*, ordering cost *Cb* and *Cv* usually are obtained from experiments, which are often imprecisely or ambiguously known to us and cannot be described by precise values. Thus, we interpret *D*, *Cb* and *Cv* as fuzzy numbers.

When making the pricing strategy, if we take the point of view from a buyer to make the profit a priority over a vendor, we can make a buyer as the leader and a vendor as the follower. By combining Formulas (1) and (2) and change *Cb* and *Cv* as fuzzy numbers, we establish a fuzzy bi-level pricing model in a supply chain as:

$$\max_{m\in M} NP_b(m) = \frac{P_{m0}\tilde{D}}{\ln(1-r_m)}[e^{H\ln(1-r_m)}-1] - P_{b0}Q\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} - \frac{F_bHP_{b0}Q}{2mn}\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} - mn\tilde{C}_b$$

$$\text{subject to } m>0 \qquad (3)$$

$$\max_{n\in N} NP_v(n) = P_{b0}Q\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} - P_{v0}mQ\frac{1-(1-r_v)^H}{1-(1-r_v)^{\frac{H}{n}}} - \frac{F_vHP_{v0}(m-1)Q}{2n}\frac{1-(1-r_v)^H}{1-(1-r_v)^{\frac{H}{n}}} - n\tilde{C}_v$$

$$\text{subject to } n>0 \qquad (4)$$

In this model, both the buyer and the vendor adjust their own controlling variables respectively, wishing to maximize their own profits, under specific constraints. The buyer is the leader, who makes decision first; and the vendor is the follower, who responses to the buyer's decision.

This model describes a nonlinear bi-level problem which is being solved by using a PSO-based algorithm.

## 4 A Particle Swarm Optimization Based Algorithm

In this section, we use the strategy adopted in the PSO method to develop a PSO-based algorithm to solve the problems defined in (3) and (4).

Figure 1 outlines the structure and process of this algorithm. We first sample the buyer-controlled variables to find some candidate choices for a buyer. Then, we introduce fuzzy coefficients by the fuzzy ranking method defined in Definition 2.1. The PSO method enables the vendor's response for every buyer's choice. Thus, a pool of strategies for both the buyer and the vendor is formed. By pushing every strategy pair towards the current best ones, the whole strategy pool is updated. Once a strategy is reached for the buyer, we use the stretching technology [21] to avoid local optimization. We repeat this procedure by a pre-defined count and reach a final decision.

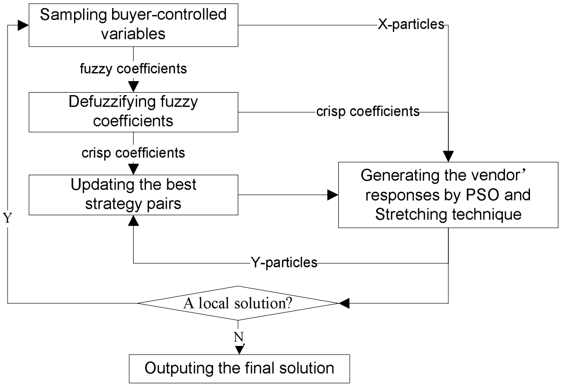

Sampling buyer-controlled variables
X-particles
fuzzy coefficients
Defuzzifying fuzzy coefficients
crisp coefficients
crisp coefficients
Updating the best strategy pairs
Generating the vendor' responses by PSO and Stretching technique
Y
Y-particles
A local solution?
N
Outputing the final solution

**Fig. 1.** The outline of the PSO based algorithm

The detailed algorithm is specified in Algorithm 1:

**Algorithm 1. A PSO based algorithm for bi-level pricing problems**

```
Step 1: Sample $N_l$ particles of $x_{ij}$, and the corresponding velocities $v_{x_{ij}}$ ;
Step 2: Initiate the buyer's loop counter $k_l = 0$;
Step 3: For the $k$-th particle, $k =1,\dots, N_l$, generate the response from the vendor;
    Step 3.1: Sample $N_f$ candidates $y_i$ and the corresponding velocities $v_{y_i}$ , $i = 1,\dots, N_f$ ;
    Step 3.2: Initiate the vendor's loop counter $k_f =0$;
    Step 3.3: By using the fuzzy ranking method defined in Definition 2.1, calculate the function value of every particle by Formula (4);
    Step 3.4: Record the best particles $p_{y_i}$ and $y^*$ from $p_{y_i}$ , $i = 1,\dots, N_f$ ;
```

Step 3.5: Update the vendor's velocities and positions using

$$v_{y_i}^{k+1} = w_f v_{y_i}^{K} + c_f r_{1l}^K (p_{y_i} - y_i^K) + c_f r_{2l}^K (y^{*K} - y_i^K)$$

$$y_{\mathrm{i}}^{K+1} = y_i^K + v_{y_i}^{k+1}$$

Step 3.6: $k_f = k_f + 1$;

Step 3.7: If $k_f \geq MaxK_f$ or the solution changes for several consecutive generations are small enough, then we use Stretching technology to obtain the global solution and go to Step 3.8. Otherwise go to Step 3.5;

Step 3.8: Output $y^*$ as the response from the vendor.

Step 4: By using the fuzzy ranking method defined in Definition 2.1, calculate the function value of every particle by Formula (3);

Step 5: Record $p_{x_{ij}}$, $x_{ij}^*$, $j = 1,\ldots, N_l$ for each $x_{ij}, j = 1,\ldots, N_l$ ;

Step 6: Update velocities and positions using

$$v_{x_{ij}}^{k+1} = w_l v_{x_{ij}}^{K} + c_l r_{1l}^K (p_{x_{ij}} - x_{ij}^K) + c_l r_{2l}^K (x_{ij}^{*K} - x_{ij}^K)$$

$$x_{ij}^{K+1} = x_{ij}^K + v_{x_{ij}}^{k+1}$$

Step 7: $k_l = k_l + 1$;

Step 8: If the sum of the differences between samples and optimal responses is smaller than $\varepsilon$ or $k_l \geq MaxK_l$ we use the Stretching technology for the current leaders' solutions to obtain the global solution.

[end]

## 5 An Example and Experiments

In this section, we illustrate the model and the algorithm developed in this study by the following numerical example and parameters:

[1] Demand rate per week, $\tilde{D} = (\underline{D}, D, \overline{D}) = (350 \text{ units}, 400 \text{ units}, 450 \text{ units})$

[2] Vendor's unit purchase cost at the initial time, *Pv0* = $4

[3] Buyer's unit purchase cost at the initial time, *Pb0* = $5

[4] Market price to the end consumer from the buyer at the initial time, *Pm0*=$6

[5] Buyer's ordering cost per order, $\tilde{Cb} = (\underline{Cb}, Cb, \overline{Cb}) = (\$25, \$30, \$35)$

[6] Vendor's ordering cost per order, $\tilde{Cv} = (\underline{Cv}, Cv, \overline{Cv}) = (\$900, \$1000, \$1100)$

[7] Buyer's holding cost per dollar per week, *Fb* = 0.004

[8] Vendor's holding cost per dollar per week, *Fv* = 0.004

[9] Time horizon considered, *H*= 52 weeks

In Table 2, we list the running results from our previous study [22], where the demand rate and the ordering cost from both the buyer and the vendor are all crisp numbers,.

**Table 2.** Summary and comparison of running results

| | *Result from [22]* | *Result from this study* |
|---|---|---|
| *m* | *16* | *16* |
| *rm* | *0.0175* | *0.0175* |
| *n* | *11* | *11* |
| *rb* | *0.0367* | *0.0367* |
| *rv* | *0.0999* | *0.0999* |
| *NPb* | *30,263* | *30,263* |
| *NPv* | *15,704* | *15,704* |
| *NP* | *46,327* | *46,327* |

From Table 2, we can see that, when the values of the coefficients are described by fuzzy numbers, the final decisions are quite similar. In this example, they are exactly the same. This is understandable due to our decision making process. When making decisions, decision makers may make small adjustment on the uncertain information, preference or circumstances. If the minor change occurs in the input, there should normally be no tremendous change to the final solution.

## 6 Conclusions

This paper establishes a fuzzy bi-level pricing model in a supply chain. A PSO based algorithm is then developed to solve the problems defined by this model. Experiment results show that the fuzzy bi-level pricing model and the PSO based algorithm can provide effective solutions to decision makers when there are uncertain information in the original parameters. Future researches and experiments can be done to explore more complex industrial applications of bi-level programming techniques in supply chain management.

## Acknowledgment

The work presented in this paper was supported by Australian Research Council (ARC) under discovery grant DP0557154.

## References

1. Yang, P., Wee, H., Yu, J.: Collaborative pricing and replenishment policy for Hi-tech industry. Journal of the Operational Research Society 58, 894–909 (2007)
2. Sern, L.C.: Present and future of supply chain in information and electronic industry. In: Supply Chain Management Conference for Electronic Industry, vol. 6 (2003)
3. Lev, B., Weiss, H.: Inventory models with cost changes. Operations Research 38, 53–63 (1990)
4. Goyal, S.: A note on inventory models with cost changes. Operations Research 40, 414–415 (1992)

5. Gascon, A.: On the finite horizon EOQ model with cost changes. Operations Research 43, 716–717 (1995)
6. Buzacott, J.: Economic order quantities with inflation. Operational Research Quarterly 26, 553 (1975)
7. Erel, E.: The effect of continuous price change in the EOQ. Omega 20, 523–527 (1992)
8. Yang, P., Wee, H.: A quick response production strategy to market demand. Production Planning & Control 12, 326–334 (2001)
9. Khouja, M., Park, S.: Optimal lot sizing under continuous price decrease. Omega 31, 539–545 (2003)
10. Zadeh, L.A.: Fuzzy sets. Information & Control 8, 338–353 (1965)
11. Lu, J., Shi, C., Zhang, G.: On bilevel multi-follower decision making: General framework and solutions. Information Science 176, 1607–1627 (2006)
12. Lu, J., Shi, C., Zhang, G.: An extended branch and bound algorithm for bilevel multi-follower decision making in a referential-uncooperative situation. International Journal of Information Technology and Decision Making 6, 371–388 (2006)
13. Yu, H., Dang, C., Wang, S.: Game Theoretical Analysis of Buy-it-now Price Auctions. International Journal of Information Technology and Decision Making 5, 557–581 (2006)
14. Hobbs, B.F., Metzler, B., Pang, J.S.: Strategic gaming analysis for electric power system: an MPEC approach. IEEE Transactions on Power System 15, 637–645 (2000)
15. Zhang, G., Lu, J.: Model and approach of fuzzy bilevel decision making for logistics planning problem. Journal of Enterprise Information Management 20, 178–197 (2007)
16. Amat, J., McCarl, B.: A representation and economic interpretation of a two-level programming problem. Journal of the Operational Research Society 32, 783–792 (1981)
17. Feng, C., Wen, C.: Bi-level and Multi-objective Model to Control Traffic Flow Into the Disaster Area Post Earthquake. Journal of the Eastern Asia Society for Transportation Studies 6, 4253–4268 (2005)
18. Gao, Y., Zhang, G., Lu, J., Gao, S.: A Bilevel Model for Railway Train Set Organizing Optimization. In: 2007 International Conference on Intelligent Systems and Knowledge Engineering, pp. 777–782 (2007)
19. Lee, E.S., Li, R.L.: Comparison of fuzzy numbers based on the probability measure of fuzzy events. Comput. Math. Appl. 15, 887–896 (1988)
20. Lu, H., Chen, W.: Self-adaptive velocity particle swarm optimization for solving constrained optimization problems. Journal of Global Optimization 41, 427–445 (2008)
21. Parsopoulos, K., Vrahatis, M.: Recent approaches to global optimization problems through Particle Swarm Optimization. Natural Computing 1, 235–306 (2002)
22. Gao, Y., Zhang, G., Lu, J., Wee, H.: A Bi-level pricing model and a Pso based algorithm in supply chain. Accepted by The 21st International Conference on Software Engineering and Knowledge Engineering (2009)

# Growing Particle Swarm Optimizers with a Population-Dependent Parameter

Chihiro Kurosu[1], Toshimichi Saito[1], and Kenya Jin'no[2]

[1] Hosei University, Tokyo, 184-8584 Japan
[2] Nippon Institute of Technology, Saitama, 345-8501 Japan

**Abstract.** This paper studies a new version of growing particle swarm optimizers. In the algorithm, a new particle is born if a particle exploring the optimum is stagnated and the swarm can grow depending on problem complexity. The particle velocity is controlled by an acceleration parameter that can attenuate depending on the number of particles and can vibrate depending on the time. The parameter plays important role to reduce the computation cost and to increase the success rate. The algorithm efficiency is confirmed by numerical experiments of typical benchmarks.

## 1 Introduction

The particle swarm optimizer (PSO) is an optimization algorithm inspired by flocking behavior of living beings [1]. In the algorithm, particles represent potential solutions and organize a swram. Referring to search history of the swarm, the particles can find the optimum solution. A variety of improved PSOs have been studied with important problems such as multi-objective optimization [4] [5], niching [2] [3], network topology [6]-[10]. Performance of the PSOs have been compared with other evolutionary computation algorithms and the results are competitive to each other in typical benchmarks [10]. The PSOs have been applied to many engineering objectives including signal processors, communication systems, multi-layer perceptrons, power systems and power electronics circuits ( see [11]-[18] and references therein).

However, the study of PSOs is still in progress. Remarkable problems still remain, for example, trapping of the entire swarm into a local optimum, suitable adjustment of algorithm parameters, increasing computation cost, and increasing inter-particle communication cost. In order to consider such problems, we have proposed the growing particle swarm optimizer (GPSO [9,10]). In the GPSO, a new particle can be born if a particle exploring the optimum is stagnated. Depending on the problem complexity, the particles can be born successively and the swarm can grow. The growing structure can be effective in escaping from a trap and finding the optimum with low computation cost. It is not natural to set common swarm size for various problems. The swarm size should change depending on property of each problem. This paper proposes an improved version

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 234–241, 2009.

of the GPSO where the particle velocity is controlled by an acceleration parameter that does not exist in our previous works. The parameter can attenuate depending on the number of particles and can vibrate depending on the time. The vibration can prevent excessive swarm convergence to a trap. The parameter plays important role to reduce the computation cost and to increase the success rate as the problem complexity increases. Performing numerical experiments for typical benchmarks, the algorithm efficiency is confirmed. These results provide basic information in order to develop flexible optimization algorithm that can adapt problem size and complexity.

## 2 Growing PSOs with Dynamic Parameter

Here we define the GPSO for a fundamental optimization problem: finding the unique minimum of an $m$-dimensional function $F(\boldsymbol{x})$ where $\boldsymbol{x} \equiv (x_1, \cdots, x_m) \in R^m$ and $F(\boldsymbol{x}) \in R$. Let $D_s \subset R^m$ denote a search space including the optimum of $F$. Each particle at discrete time $t$ is characterized by its position $\boldsymbol{x}_i \equiv (x_{i1}, \cdots, x_{im}) \in R^m$, velocity $\boldsymbol{v}_i \equiv (v_{i1}, \cdots, v_{im}) \in R^m$, and a counter $c_i(t)$. The position $\boldsymbol{x}_i$ corresponds to a potential solution and each particle moves to minimize $F(\boldsymbol{x}_i)$. The personal best (*pbest*) that is the best value of the $F(\boldsymbol{x}_i)$ in the past search process. The optimum at each time step is given by the global best (*gbest*) that is the best of *pbest* for all the particles. The counter $c_i(t)$ inspects time-invariance of the $pbest_i$ and is used for controlling the number of particles $N(t)$. Although there exists a variety of the swarm topology [10], we adapt the complete graph for simplicity. Note that $N(t)$ can increase and the swarm can grow in the search process. The algorithm is defined as the following:

**Step 1** (Initialization): Let $t = 0$ and let initial particle numbers $N(t)$ be given. The particle positions $\boldsymbol{x}_i(t)$, $i = 1 \sim N(t)$, are assigned randomly in the search space $D_s$. The velocity, counter value, pbest and gbest are all initialized: $c_i(t) = 0$, $\boldsymbol{v}_i = \boldsymbol{0}$, $pbest_i = F(\boldsymbol{x}_i(t))$ and $gbest \leq pbest_i$.

**Step 2** (Renewal of particle positions):

$$\begin{aligned} &\boldsymbol{x}_i(t) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t) \\ &\boldsymbol{v}_i(t) = w(N(t),t)\boldsymbol{v}_i(t) + \rho_1(\boldsymbol{x}_{pbest_i} - \boldsymbol{x}_i(t)) + \rho_2(\boldsymbol{x}_{gbest} - \boldsymbol{x}_i(t)) \end{aligned} \tag{1}$$

where $i = 1 \sim N(t)$. The parameters $\rho_1$ and $\rho_2$ are selected randomly from a uniform distribution on $[0, 2]$. The acceleration parameter $w(N(t), t)$ can attenuate depending on the population $N(t)$ and can vibrate depending on $t$:

$$w(N(t),t) = (-1)^t w_0 A^{N(t)-N(0)} \tag{2}$$

where $0 < w_0 < 1$ is an initial value and $0 < A < 1$ is an attenuation parameter. As suggested in Section 1, Eq. (2) is a key term in order to improve the algorithm efficiency.

**Step 3** (Update of pbest): The current value of $F(x_i(t))$ is compared with $pbest_i$ and the counter value is increased if $pbest_i$ is not changed:

$$\begin{aligned}&pbest_i = F(\boldsymbol{x}_i(t)),\ \boldsymbol{x}_{pbext_i} = \boldsymbol{x}_i(t) \text{ if } F(\boldsymbol{x}_i(t)) < pbest_i,\\ &c_i(t) = \begin{cases} c_i(t)+1 \text{ if } F(\boldsymbol{x}_i(t)) \geq pbest_i \\ 0 \qquad\quad\ \text{if } F(\boldsymbol{x}_i(t)) < pbest_i \end{cases}, \qquad i = 1 \sim N(t)\end{aligned} \tag{3}$$

**Step 4** (Birth of particles): If a counter value $c_i(t)$ exceeds the threshold $T_c$ then a new particle is born and is located randomly in $D_s$ where $i = 1 \sim N(t)$. Let $N(t) = N(t) + N_b$ if the number of the born particles is $N_b$. The new particles can be born until $N(t)$ reaches the maximum population limit $N_{max}$.

**Step 5** (Update of *gbest*):

$$gbest = pbest_i \text{ and } \boldsymbol{x}_{gbest} = \boldsymbol{x}_{pbest_i} \text{ if } pbest_i < gbest, \quad i = 1 \sim N(t). \tag{4}$$

If $F(\boldsymbol{x}_{gbest})$ satisfies the criterion then the algorithm is terminated.

**Step 6**: $t = t+1$, return to Step 2 and repeat until the maximum time limit $t_{max}$.

Note again that this GPSO has a novel acceleration parameter $W(N(t), t)$ of population-dependent attenuation with time-dependent vibration in Step 2. Hereafter this algorithm is referred to as PATV for convenience.

## 3 Numerical Experiments

In order to confirm the algorithm efficiency, we have applied the PATV to two typical benchmarks. The first one is the Levy function:

$$\begin{gathered} F_L(\boldsymbol{x}) = \sum_{i=1}^{m-1} [(y_i-1)^2(1+10\sin^2(\pi y_i)] \\ +(y_m-1)^2(1+10\sin^2(2\pi y_m)+\sin^2(\pi y_1) \\ y_i \equiv 1+(x_i-1)/4, \quad \min(F_L(\boldsymbol{x})) = F_L(1, ..., 1) = 0 \end{gathered} \tag{5}$$

The landscape is illustrated in Fig. 1. We have judged that the optimum is obtained if the criterion $F_L(\boldsymbol{x}) \leq 10^{-3}$ is satisfied. Such a criterion is given by referring to several experiments including [10]. At $t = 0$, we have assigned $N(0) = 3$ particles in the 2-dimensional search space $D_s = \{x| -10 \leq x_i \leq 10, i = 1 \sim m\}$, $m = 2$. Figs. 2 and 3 show the search process where the swarm grows and the value of $F_L$ decreases to attain the criterion.

The second one is the Schwefel function:

$$\begin{aligned} &F_S(\boldsymbol{x}) = 418.9829 - (\sum_{i=1}^{m} x_i \sin(\sqrt{|x_i|})/m \\ &\min(F_S(x)) = F_S(420.968750, \ldots, 420.968750) = 0 \end{aligned} \tag{6}$$

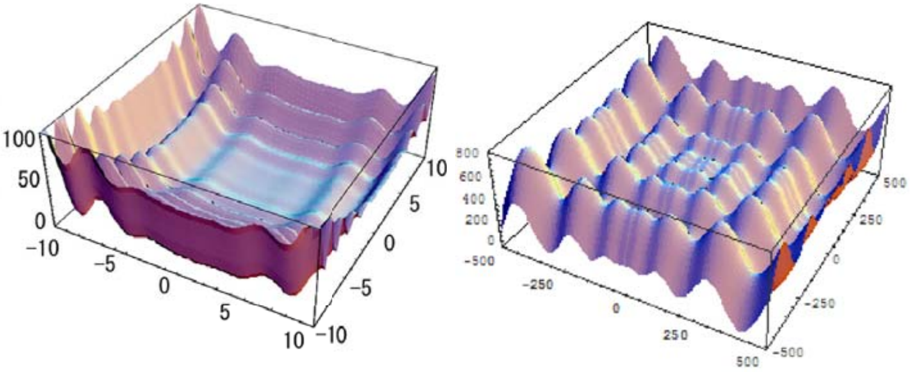


**Fig. 1.** Landscape of Levy (left) and Schwefel (right) functions

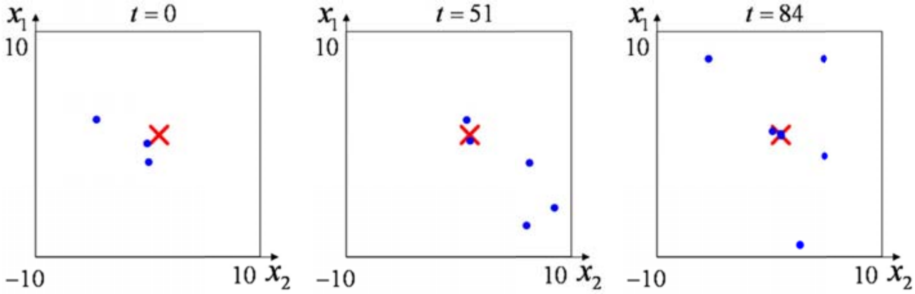


**Fig. 2.** Snapshot of growing swarm for the Levy function (PATV). The cross denotes the optimum.

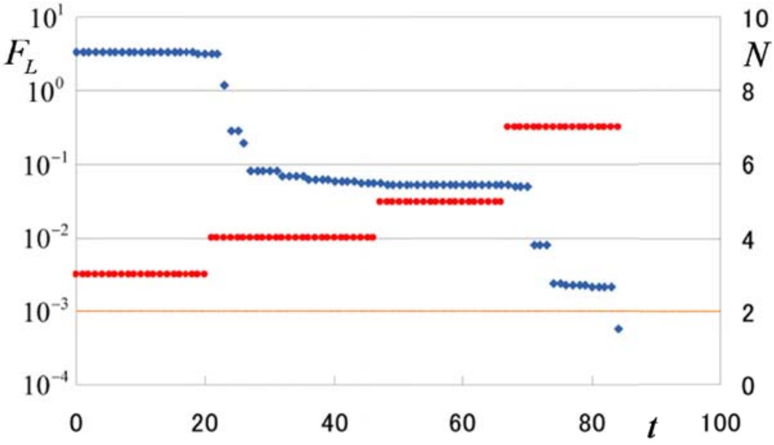


**Fig. 3.** Search process of Levy function (PATV). Bule = *gbest* for the left scale, Red = # particles for the right scale and the criterion = $10^{-3}$.

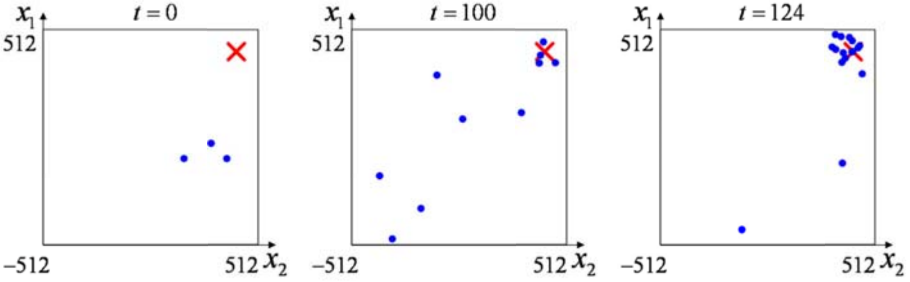


**Fig. 4.** Snapshot of growing swarm for the Schwefel function (PATV). The cross denotes the optimum.

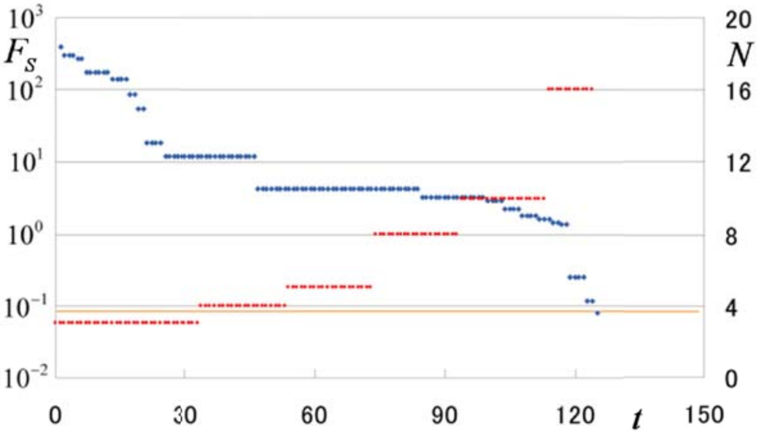


**Fig. 5.** Search process of Schwefel function (PATV). Bule = *gbest* for the left scale, Red = # particles for the right scale and the criterion = $8.29 \times 10^{-2}$.

We have judged that the optimum is obtained if the criterion $F_S(\boldsymbol{x}) \leq 8.29 \times 10^{-2}$ is satisfied. At $t = 0$, we have assigned $N(0) = 3$ particles in the 2-dimensional search space $D_s = \{x | -512 \leq x_i \leq 512, i = 1 \sim m\}$, $m = 2$. Figs. 4 and 5 show the growing swarm and search process. This function is complicated than the Levi function, however, the PATV can find the optimum.

We have performed 100 trials for different initial conditions and random parameters $\rho_1$ and $rho_2$. The results are evaluated in the following three measures and are summarized in the first column of the Tables 1 and 2.

SFR: The successful rate of the criterion attainment. It measures search ability.

#ITL(AV/SD): The number of iterations to attain the criterion (Average/ Standard deviation). It measures the search speed.

**Table 1.** Basic performance of Levy function

| Metod | SFR[%] | #ITL(AV) | #ITL(SD) | #PCL(AV) | #PCL(SD) |
|---|---|---|---|---|---|
| PATV | 100 | 125.3 | 26.8 | 18.6 | 9.0 |
| PA | 100 | 113.5 | 52.8 | 9.3 | 6.6 |
| TATV | 100 | 145.3 | 41.2 | 34.3 | 23.7 |
| TA | 97 | 139.6 | 120.2 | 8.3 | 5.4 |
| CPSO | 100 | 28.5 | 12.3 | 15 | n/a |

**Table 2.** Basic performance of Schwefel function

| Metod | SFR[%] | #ITL(AV) | #ITL(SD) | #PCL(AV) | #PCL(SD) |
|---|---|---|---|---|---|
| PATV | 100 | 150.1 | 33.0 | 25.5 | 15.4 |
| PA | 94 | 239.3 | 115.5 | 52.4 | 36.6 |
| TATV | 100 | 187.0 | 79.7 | 51.3 | 31.0 |
| TA | 79 | 295.7 | 167.7 | 48.3 | 33.7 |
| CPSO | 61 | 37.1 | 107.1 | 50 | n/a |

#PCL(AV/AD): The average number of particles at the criterion attainment (Average/Standard deviation). It measures the swarm scale.

In order to evaluate these results, we have repeated similar numerical experiments for other four cases:

PA: Population-dependent attenuation without vibration. It is given by replacing $w$ in Step 2 with

$$w(N(t)) = w_0 A^{N(t)-N(0)} \tag{7}$$

TATV: Time-dependent attenuation with time-dependent vibration. It is given by replacing $w$ in Step 2 with

$$w(t) = (-1)^t (w_0 - rt) \tag{8}$$

TA: Time-dependent attenuation without vibration. It is given by replacing $w$ in Step 2 with

$$w(t) = w_0 - rt \tag{9}$$

CPSO: PSO without growing. In this algorithm, the number of particle is fixed $N(t) = N_c$ and $w$ in Step 2 is given by Eq. (9).

The results are summarized in Tables 1 and 2:

- The PATV, PA, TATV and CPSO always find the optimum of the Levy function. The PATV and TATV always the optimum of the Schwefel function, however, the CPSO and TA are hard to find it.

- The CPSO can find the optimum of the Levy with smaller #ITL than others, however, SFR for the Schwefel is 61%. It suggests that the CPSO is effective for relatively simple problems but can not work in complicated problems.
- In the Schwefel, the PATV can find the optimum with smaller #ITE and #PCL than PA and TATV. TA and CPSO sometimes fail to find the optimum. It suggests that interaction between population-dependent attenuation and time-dependent vibration is necessary to realize faster search and lower computation cost (smaller #ITL and #PCL) in relatively complex problems.
- We suppose that population-dependent attenuation suppresses unnecessary divergence and time-dependent vibration realizes precise search. PA (population-dependent attenuation only), TA (time-dependent attenuation only) and CPSO are insufficient.
- The growing structure can set a swarm size which is suitable for a given problem. Note that the territory per particle is given by the #PCL.

## 4 Conclusions

An improved version of the GPSO with an acceleration parameter is studied in this paper. The parameter can attenuate depending on the number of particles, can vibrate depending on the time. The parameter controls the particle velocity flexibly. Performing basic numerical experiments, we have suggest that the algorithm is effective to realize faster and lower cost optimization. This paper provides a first step to establish PSOs in which parameters can be adjusted adequately and automatically. Future problems are many, including the following : Analysis of search process and role of parameters, Application of a variety of swarm topology. Application to wider class of benchmarks and to practical problems such as design in power electronics [18].

## Acknowledgement

This work is supported in part by JSPS KAKENHI#21500223.

## References

1. Engelbrecht, A.P.: Fundamentals of computational swarm intelligence. Wiley, Chichester (2005)
2. Parsopoulos, K.E., Vrahatis, M.N.: On the computation of all global minimizers through particle swarm optimization. IEEE Trans. Evol. Comput. 8(3), 211–224 (2004)
3. Schoeman, I.L., Engelbrecht, A.P.: Using vector operations to identify niches for particle swarm optimization. In: Conf. Cybern. Intel. Systs., pp. 361–366 (2004)
4. Parrott, D., Li, X.: Locating and tracking multiple dynamic optima by a particle swarm model using speciation. IEEE Trans. Evol. Comput. 10(4), 440–458 (2006)
5. Chow, C.K., Yuen, S.Y.: A non-revisiting particle swarm optimization. In: Congr. Evol. Comput., pp. 1879–1885 (2008)

6. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. In: Congr. Evol. Comput., pp. 1671–1676 (2002)
7. Mendes, R., Kennedy, J., Neves, J.: The fully informed particle swarm: simpler, maybe better. IEEE Trans. Evol. Comput. 8(3), 204–210 (2004)
8. Godoy, A., Von Zuben, F.J.: A Complex Neighborhood Based Particle Swarm Optimization. In: Congr. Evol. Comput., pp. 720–727 (2009)
9. Miyagawa, E., Saito, T.: Particle Swarm Optimizers with Grow-and-reduce Structure. In: Proc. Congr. Evol. Comput., pp. 3975–3980 (2008)
10. Miyagawa, E., Saito, T.: Particle Swarm Optimizers with Growing Tree Topology. IEICE Trans. Fundamentals E92-A (accepted, 2009)
11. Hsieh, S.-T., Sun, T.-Y., Lin, C.-L., Liu, C.-C.: Effective learning rate adjustment of blind source separation based on an improved particle swarm optimizer. IEEE Trans. Evol. Comput. 12(2), 242–251 (2008)
12. Jatmiko, W., Sekiyama, K., Fukuda, T.: A PSO-based mobile sensor network for odor source localization in dynamic environment: theory, simulation and measurement. In: Congr. Evol. Comput., pp. 3781–3788 (2006)
13. Teixeira, F., Romariz, A.: Digital filter arbitrary magnitude and phase approximations-statistical analysis applied to a stochastic-based optimization approach. In: Congr. Evol. Comput., pp. 4089–4096 (2008)
14. Oliveira, L.D., Abrao, T., Jeszensky, P.J.E., Casadevall, F.: Particle swarm optimization assisted multiuser detector for M-QAMDS/CDMA systems. In: Swarm Intelligence Symposium (2008)
15. Valle, V., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.-C., Harley, R.G.: Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Trans. Evol. Comput. 12(2), 171–195 (2008)
16. Rakitianskaia, A., Engelbrecht, A.P.: Training Neural Networks with PSO in Dynamic Environments. In: Congr. Evol. Comput., pp. 667–673 (2009)
17. Garro, B.A., Sossa, H., Vazquez, R.A.: Design of Artificial Neural Networks using a Modified Particle Swarm Optimization Algorithm. In: Int'l Joint Conf. Neural Netw., pp. 938–945 (2009)
18. Ono, K., Saito, T.: Application of Particle Swarm Optimizers to Two-Objective Problems in Design of Switching Inverters. In: Int'l Joint Conf. Neural Netw., pp. 2353–2357 (2009)

# An Efficient Feature Selection Using Ant Colony Optimization Algorithm

Md. Monirul Kabir[1], Md. Shahjahan[2], and Kazuyuki Murase[1]

[1] Department of Human and Artificial Intelligence Systems
University of Fukui, Bunkyo 3-9-1, Fukui 910-8507, Japan
{kabir,murase}@synapse.his.fukui-u.ac.jp
[2] Department of Electrical and Electronic Engineering
Khulna University of Engineering and Technology, Khulna 9203, Bangladesh
jahan@mail.kuet.ac.bd

**Abstract.** This paper presents an efficient feature selection algorithm by utilizing the strategy of ant colony optimization, called as ACOFS. Initially, ACOFS uses a modified framework to guide the ants in the right directions while constructing the graph (subset) paths. In the subsequent part, a set of new modified pheromone update rules as well as a set of new modified estimation of heuristic information for features are introduced. The effect of such modifications ultimately assists ants to generate salient feature subsets with reduced size. We evaluate the performance of ACOFS on four real-world benchmark datasets. The experimental results show that ACOFS has a remarkable capability to generate reduced size subsets of salient features with yielding significant classification accuracies.

**Keywords:** Feature selection, pheromone update, heuristic information, neural network, classification accuracy.

## 1 Introduction

Feature selection (FS) has a crucial role in data mining. The aim of FS is to reduce the dimension of original feature set by identifying the spurious features properly which ultimately provides the best performances under some classification datasets. However, searching of spurious features by their dependency on the classification algorithm can be divided into two categories, such as: filters and wrappers [1], [2].

In accordance with the researchers, FS is basically a search process. In finding the salient features, search can be categorized by forward search [3] and backward search [4]. Furthermore, search can also be carried out by genetic algorithm [5] or genetic programming [6], called genetic search. Apart from these, there are also some other search techniques in FS which are: Tabu seach, Simulated annealing, and so on. Ant colony optimization (ACO) is an algorithm that is inspired by social behavior of ant colonies and has already been used in many combinatorial optimization problems for optimum solutions. For instance, ACO in TSP [7], QAP [8], routing in telecommunication networks, graph coloring problem, and so on. In case of FS problems, recently a number of approaches have been introduced [9]-[15].

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 242–252, 2009.

In this paper, we propose an efficient FS algorithm, called ant colony optimization based FS algorithm (ACOFS). The main focus of this algorithm is to modify the inner operation of the ACO algorithm according to the activity of wrapper tool rather than filter tool. It is reasonable in the sense that the performance of wrapper model is outperformed filter model [2][3]. The proposed technique adapts a set of new modified pheromone update rules and heuristic information estimations based on the evaluation of constructed subsets by means of trained neural network (NN) classifier. Thus ACOFS differs from previous works (e.g., [9]-[15]) on three vital aspects.

First, ACOFS emphasizes on deciding the size of the subset in a restricted random fashion while ants attempt to traverse upon the node (or, feature) space to construct a path (or, subset). This subject is represented by means of a modified framework mentioned in Section 2.1. Such approach is quite different from the existing works [9][10][14] in view of guiding ants to be used the forward selection strategy (FSS) during subset constructions (SCs). It is needed to say here that FSS always suffers by two problems: (a) a suitable stopping criterion is necessary to stop the SC; (b) proper guidance is required for adding or deleting features during SC. However, to solve the above problems, a strategy of fixed size is incorporated in [12][13][15] which is augmented by a constant rate or randomly chosen, but needs suitable boundary in between upper and lower limit. Thus deciding the subset size randomly within a constrained limit may be a novel approach in this paper during SCs.

Second, ACOFS uses a conventional probabilistic transition rule [9][14] in selecting features during SCs. Since, initially all the features in a given dataset possess equal pheromone value and heuristic value, such transition rule shows the random behavior. For this, ACOFS imposes an extra constraint upon the ants during feature SC. It is reasonable because of three aims which are: (a) all original features may be included, (b) entire features may get chance to be evaluated themselves, and (c) ACOFS can be free from the premature convergence. According to our sense, such amendment may lead to achieve a better result for the subsequent iterations. No other approaches [9][13][14] consider such technique in case of random manner.

Third, there are two new modified sets of rules are proposed in ACOFS which are pheromone updating and heuristic information estimating, respectively. Because of appearing the random behavior in selecting features during SCs initially, the above rules are proposed which afford some advantages, such as: (a) providing the correct update information to the features, (b) providing the effective balance between exploitation and exploration of ants. Mostly, the proposed rules are dependent on the outcomes of trained NN. The existing FS approaches do not consider such modifications in designing those rules (e.g. [9][10][13][14]).

The remainder of this paper is organized as follows. The details of ACOFS are discussed in Section 2. Experimental results, comparison to other methods, and the brief computation of computational complexity are reported in Section 3. Short conclusions with few remarks are given in Section 4.

## 2 Proposed ACOFS

In this proposed approach, we use NN training classifier for evaluating the constructed subsets. Involving any heuristic strategy in the classifier part also brings great impact in generating the salient subsets. It is noted that, most of the ACO based FS algorithms have less attention for the betterment of the activity of classifiers. However, a strategy of determining the required number of hidden neuron in the hidden layer of NN, called constructive strategy is induced in ACOFS. Details of this strategy can be seen in [16].

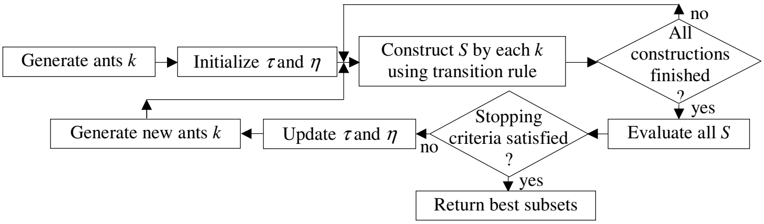

Generate ants $k$
Initialize $\tau$ and $\eta$
Construct $S$ by each $k$ using transition rule
All constructions finished ?
no
yes
Evaluate all $S$
Stopping criteria satisfied ?
no
yes
Update $\tau$ and $\eta$
Generate new ants $k$
Return best subsets

**Fig. 1.** Flow chart of ACOFS

The steps of ACOFS can be described by the flowchart shown in Fig. 1, which are explained further as follows.

Step 1) Initially generate a set of artificial ants $k$ and the number of $k$ must be equal to the number of original features $n$, i.e., $k \equiv n$.

Step 2) Initialize the level of pheromone trails $\tau$ and the heuristic information $\eta$ of all features $n$ by setting the equal value.

Step 3) Follow the scheme mentioned in Section 2.1 to be decided the subset size $r$ in feature SC initially. After that, follow the conventional probabilistic transition rule [9][14] to select the features that is mentioned as follows,

$$P_i^k(t) = \begin{cases} \dfrac{[\tau_i(t)]^\alpha * [\eta_i(t)]^\beta}{\sum\limits_{u \in j^k} [\tau_u(t)]^\alpha * [\eta_u(t)]^\beta} & if\ i \in j^k \\ 0 & otherwise \end{cases} \tag{1}$$

where, $j^k$ is the set of feasible features that can be added to the partial solution. $\tau_i$ and $\eta_i$ are the pheromone value and heuristic value associated with feature $i$. $\alpha$ and $\beta$ are the parameter of their relative importance. It should be noted that as initially all features contain equal value of $\tau$ and $\eta$, the Eq. (1) shows the random behavior. Thus to get the fair selection of features we impose one constraint upon the ants $k$ during SCs at $t$ iteration as,

$$S^k(t) \neq S^{k+1}(t) \qquad k = 1,2,3,.........,n \tag{2}$$

Step 4) Check the construction progresses of $S$ to determine whether it is finished or not. If finished by all ants $k$ then continue, otherwise, go to Step 3).

Step 5) Evaluate the subsets $S^k(t)$ using NN during training. The classification accuracy ($CA$) of those subsets can be calculated as,

$$\gamma(S^k(t)) = 100 * (1 - TER) \tag{3}$$

Here, *TER* refers to the error rate of the NN in testing set.

Step 6) Check whether the algorithm has executed a certain number of iterations *T* or not. If executed then the FS process is stopped and the best salient feature subset can be obtained according to Section 2.2, otherwise, continue.

Step 7) Update $\tau$ and $\eta$ of each feature *i* by following Section 2.3 and 2.4, respectively.

Step 8) Generate a new set of artificial ants *k* and go to Step 3) to be iterated the above mentioned similar procedures.

It is now clear that the basic steps behind ACOFS is nearly similar and straightforward compared to the existing algorithms (e.g., [9-15]) but differ in terms of some modifications integrated here. The following sections give further more details about the different components of ACOFS.

## 2.1 Determination of Subset Size

Unlike other algorithms, ACOFS uses a new modified determination scheme for subset size *r* from [6], called *restricted random scheme*. The aim is to provide the value of *r* for the individual ant *k* in a reasonable boundary. For that reason, the proposed scheme works under a restricted area which is described as follows.

Firstly, the probabilistic value of *r* in a bounded region can be defined as,

$$P(r) = \frac{n-r}{\sum_{i=1}^{l}(n-i)} \qquad where,\ 2 \le r \le \delta\, and\, l \in n-r \tag{4}$$

Here, $\delta$ is define by $\mu$ of *n*. Specifically, $\mu$ is a user specified parameter and its value is set up to [0.1, 0.6] depending on the number of *n* of different given datasets.

Secondly, ACOFS arranges all the possible values of P(*r*) to the conventional "roulette-wheel selection" scheme [17] to achieve the size of the subsets consistently.

Finally, deciding the suitable value of $\mu$ in Eq. (4) is a difficult issue. In ACOFS, the salient subset is being generated by a predefined range of its size which is in between 2 to 12. In most of the cases, whether the dimensionality of a given dataset is medium or high, ACOFS tries to generate subsets within this range. Thus, on basis of such assumption the value of $\mu$ is decided.

## 2.2 Determination of Salient Feature Subset

In this paper, unlike others, the determination of the salient feature subset is suggested basically by two ways: i) determining *local best* subset $S^l$, ii) determining *global best* subset $S^g$. Particularly, in every iteration *t* where *t*∈ 1,2,3,…..,*T*, ACOFS first computes $S^l(t)$ according to $Max(\gamma(S^k(t)))$. Then, $\gamma(S^l(t))$ is checked whether it is greater than the current $\gamma(S^g)$ where initially $\gamma(S^g) \equiv 0$. If so then $S^l(t)$ replaces $S^g$ and associated all related components are replaced as the global components. The similar process is also happened while $\gamma(S^l(t))$ is equal to the current $\gamma(S^g)$ but

$| S^l(t) | < | S^g |$. Furthermore, if $\gamma(S^l(t)) \equiv \gamma(S^g)$ and $| S^l(t) | \equiv | S^g |$ then it requires one additional computation, i.e., information gain ($\lambda$) of the features associated with the both subsets. The subset which having higher computed value of $\lambda$ regarded as $S^g$ for the current iteration *t*. This is because, according to the statistical analysis, best feature is the one which has highest $\lambda$ [18]. Thus, to find out the salient feature subset globally the above-mentioned process is repeated until *t* is equal to *T*.

### 2.3 Pheromone Update Rule

Pheromone updating is a vital part for working the ACO algorithm suitably. In course of SCs, there are two kinds of phenomena usually happened which are: random and probabilistic. On basis of these phenomena, a set of two modified pheromone update rules are introduced from the original one mentioned in [9] [19] as follows,

i) *Random case*: A modified rule is presented in Eq. (5). Here the second part (local updating) is responsible for the modification from the original formula where $m_i$ is used to divide. The aim of such modification is to provide an equal level of pheromone to the feature *i* where $i \in n$.

$$\tau_i(t+1) = (1-\rho) * \tau_i(t) + \frac{1}{m_i} \sum_{k=1}^{n} \Delta\tau_i^k(t) + e * \Delta\tau_i^g(t) \tag{5}$$

$$where,\ \Delta\tau_i^k(t) = \begin{cases} \gamma(S^k(t)) & if\ i \in S^k(t) \\ 0 & otherwise \end{cases},\quad \Delta\tau_i^g(t) = \begin{cases} \gamma(S^l(t)) & if\ i \in S^l(t) \\ 0 & otherwise \end{cases} \tag{6}$$

Here, $m_i$, $\rho$ and *e* refer to the number of count for *i* that is associated with $S^k(t)$ in the current iteration, pheromone decay and elitist parameter, respectively. Furthermore, $\Delta\tau^k$ and $\Delta\tau^g$ are the amount of local updating and global updating, respectively.

ii) *Probabilistic case*: The modified rule is shown in Eq. (7) where the modification is in the third part (global updating) of the original form.

$$\tau_i(t+1) = (1-\rho) * \tau_i(t) + \sum_{k=1}^{n} \Delta\tau_i^k(t) + e * \Delta\tau_i^g(t) \tag{7}$$

Here, $\Delta\tau_i^k(t)$ and $\Delta\tau_i^g(t)$ can be defined by the similar way as mentioned in Eq. (6). In addition, one additional caution is incorporated in the above updating rule at the time of adding $\Delta\tau_i^g(t)$ to the particular feature *i*. It means that, if $\gamma(S^l(t)) < \max(\gamma(S^l(t')), \varepsilon)$ where *t* is the current iteration which can be $\in 2,3,\ldots,T$. $\varepsilon$ refers to the number of *CAs* of those associated subsets that maintain $| S^l(t') | \equiv | S^l(t) |$ and $t' \in (t\text{-}1),(t\text{-}2),\ldots.,1$. Thus, a number of feature $i'$ is ignored to get $\Delta\tau^g$. Here, $i' \in S^l(t')$ but $\notin S^{l*}$ where $S^{l*}$ provides $\max(\gamma(S^l(t'-\tau)), \varepsilon)$.

It is worth mentioning that the aim of such above restriction is to provide the global updating to those features only that are really significant. It is due to the fact that, global updating has a vital role in selecting the salient features in ACOFS.

### 2.4 Estimation of Heuristic Information

Generally, the representation of heuristic value is the attractiveness of features and depends on the dependency degree. Indeed, it is necessary to limit the value of $\eta$, otherwise, the algorithm may be too greedy and the better solution may not come [14].

In this paper, we propose a set of new heuristic information estimation rule depending on two basic phenomena in SC. Both rules are mainly developed from the results of subset evaluation which is indeed more effective than statistical investigation of the feature set.

i) *Random case*: The estimation of $\eta$ for the feature *i* can be performed according to the Eq. (8).

$$\eta_i = \frac{1}{m_i} \sum_{k=1}^{n} \gamma_a(S^k(t)) * (1 + \phi * e^{-\frac{|S^k(t)|}{n}}) \qquad \textit{if } i \in S^k(t) \tag{8}$$

ii) *Probabilistic case*: $\eta$ is estimated according to Eq. (9).

$$\eta_i = m_i * \varphi_i * \omega * \lambda_i(p) * \sum_{k=1}^{n} \gamma_a(S^k(t)) * (1 + \phi * e^{-\frac{|S^k(t)|}{n}}) \qquad \textit{if } i \in S^k(t) \tag{9}$$

Here, $\varphi_i$ refers to the number of count for the particular selected feature *i* that is a part of $S^k(t'')$. Here, $t''$ implies the whole completed iterations. However, the aim of multiplying $m_i$ and $\varphi_i$ in Eq. (9) is to combine the importance of *i* in terms of locally and globally. Conversely, the computation of information gain $\lambda$ [18] of features is incorporated here because of making diversity among the different features in terms of $\eta$ as well as removing the greedy manner of the specific features in feature SC. Finally, one additional exponential term is multiplied in Eq. (8) and (9). The aim of such incorporation is to provide the higher value of $\eta$ to those features that are associated with the reduced subsets and provided better performances.

## 3 Experimental Setup

In order to evaluate the performance of ACOFS, four real-world benchmark datasets [20] were used. The used datasets are: breast cancer (BCR), glass (GLS), thyroid (THY), and sonar (SNR). The characteristics of these datasets and their partitions are shown in Table 1. Each experiment was performed 20 times and the presented results are the average of these 20 runs. The performance of ACOFS was evaluated in terms of the number of selected features ($n_s$) as well as classification accuracy (*CA*). All experiments were done in Pentium-core 2 duo, 2.66 GHz personal computer.

In this study, in order to achieve better results, we set $T$=20, $\alpha$=2, $\beta$=3, $\tau$=0.5, $\eta$=0.1, $\rho$=0.4, $\omega$=0.1, and $\phi$=0.1. There were also some other parameters used in NN training while the constructed subsets were evaluated. The initial connection weights for an NN were randomly set to [-1.0, 1.0]. The learning rate and the momentum term were set to [0.1, 0.2] and [0.5, 0.9], respectively. The number of partial training epochs was chosen between [10, 70]. The training was conducted by the well-known

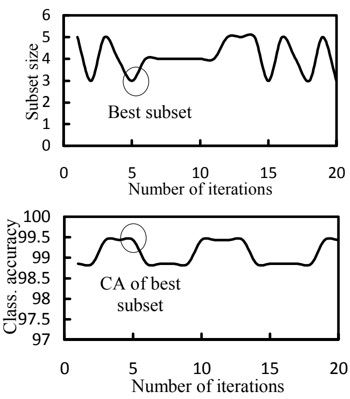


**Fig. 2.** Finding best subset of cancer dataset for a single run

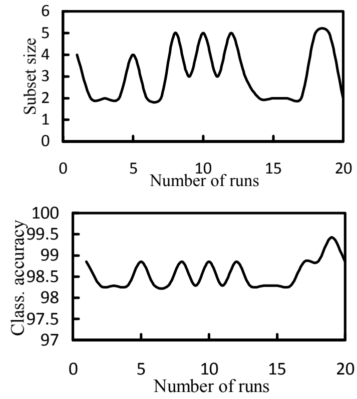


**Fig. 3.** Variation of best solutions for cancer dataset in different runs

BP algorithm [21]. We conducted one additional set of experiments to investigate the performance of the all original features using trained NN.

### 3.1 Experimental Results

The results presented in Table 2 were found by applying ACOFS on four real-world datasets. The *CA* in the table refers to the percentage of correct classification by the trained NN on the testing set. It is seen that ACOFS could select a small number of $n_s$ from the original feature set. For example, for the thyroid dataset, ACOFS selected 2.85 features on average from a set of 21 features. This indicates that ACOFS could find salient features in a reduced way. The positive effect of a small number of features can also be seen when we look the *CA*. For example, the sonar dataset, the *CA* of all features was 75.49%, when it was 86.44% with 6.65 features. ACOFS exhibits good results for other datasets as well.

Furthermore, the use of $n_s$ cause a small standard deviation (*SD*) as presented in the Table 2 for each entry. The low *SDs* imply robustness of ACOFS. Robustness is the consistency of an algorithm under different initial condition.

In order to observe how the subset solution of ACOFS progresses, Fig. 2 shows the scenery of cancer dataset in a single run how the best solution was identified. It is seen that the local best performances is varied with varying the local best subsets. There are also several points where the *CAs* are maximized but the best solution is selected (indicated by circle) in that particular run by considering the size of subset which is reduced much. On the other hand, Fig. 3 shows the variation between the subset size and *CAs* in total 20 runs. Thus, from the above discussion it can be assumed that the performance of a subset is roughly dependent on its size.

### 3.2 Comparison with Other Works

The obtained results of ACOFS on four benchmark datasets are compared with the results of different FS algorithms. Three well-known algorithms ACOFS$_S$[12], GPFS[6], and MLPFS[2] are chosen for comparison. The results are summarized in Table 3. Since, the different algorithms were evaluated in different experimental setups, for this, we cannot compare the results of ACOFS completely with other algorithms until all experiments are performed in the same setup.

Table 3 shows the comparisons between ACOFS and other algorithms on basis of average percentage of *CAs* and average number of $n_s$. Observing this Table, the comparative analysis for four datasets and other algorithms are stated below.

*Cancer*: The overall performance of ACOFS in terms of $n_s$ and *CA* is much better than all other algorithms except for one case. In this case, the number of $n_s$ in ACOFS is comparable to that of GPFS.

*Glass*: ACOFS has shown a remarkable performance in all cases comparing to MLPFS.

*Thyroid*: In terms of number of $n_s$, ACOFS achieved a few number of features comparing to ACOFS$_S$ while in case of *CA*, it also achieved better results.

*Sonar*: In terms of number of $n_s$, ACOFS achieved a few number of salient features comparing to GPFS and MLPFS. In contrast, ACOFS performed better in terms of *CA* comparing to the above two algorithms.

### 3.3 Computational Complexity

The study of computational complexity helps to understand the actual computational cost (CC) of an algorithm. For this, we are inspired to compute the CC of ACOFS briefly according to the form of big-O notation. In ACOFS, there are five vital steps: (i) subset construction, (ii) subset evaluation, (iii) stopping criterion, (iv) salient subset determination, and (v) updating pheromone and heuristic value.

*i) Subset Construction*: In this work, we use two kinds of computation for the SC because of two different kinds of phenomena exhibited by Eq. (1). For the random case, the total computational cost takes $O(k \times r)+O(k \times r^2)$ operations for $k$ ants. In terms of probabilistic case, ACOFS uses the Eq. (1) for selecting features in SC which shows the total $O(k \times r \times q)$ operations. Here, $q$ equals to $(T\text{-}1)$. In practice, the cost of generating $r$ is negligible. In contrast, the operations $O(k \times r)+O(k \times r^2)$ are performed only for once which can also be ignored.

**Table 1.** Characteristics of datasets

| Datasets | Features | Classes | Examples | Partition sets | | |
|---|---|---|---|---|---|---|
| | | | | Training | Validation | Testing |
| BCR | 9 | 2 | 699 | 349 | 175 | 175 |
| GLS | 9 | 6 | 214 | 108 | 53 | 53 |
| THY | 21 | 3 | 7200 | 3600 | 1800 | 1800 |
| SNR | 60 | 2 | 208 | 104 | 52 | 52 |

**Table 2.** Average results of BCR, GLS, THY, and SNR datasets. SD refers standard deviation.

| Datasets | Avg. results with all features | | Avg. results with selected features | | | |
|---|---|---|---|---|---|---|
| | CA (%) | SD | No. of features | SD | CA (%) | SD |
| BCR | 97.60 | 0.003 | 3.10 | 1.26 | 98.57 | 0.33 |
| GLS | 68.59 | 0.065 | 3.10 | 1.17 | 80.75 | 1.27 |
| THY | 97.99 | 0.002 | 2.85 | 1.15 | 98.36 | 0.13 |
| SNR | 75.49 | 0.082 | 6.65 | 1.93 | 86.44 | 0.73 |

**Table 3.** Comparisons between ACOFS, ACOFS$_S$[12], GPFS[6], and MLPFS[2]

| Datasets | | Comparisons | | | |
|---|---|---|---|---|---|
| | | ACOFS | ACOFS$_S$ | GPFS | MLPFS |
| BCR | No. of features | 3.10 | 12.00 | 2.23 | 8.00 |
| | Class. acc. (%) | 98.57 | 95.57 | 96.84 | 89.40 |
| GLS | No. of features | 3.10 | -- | -- | 8.00 |
| | Class. acc. (%) | 80.75 | -- | -- | 44.10 |
| THY | No. of features | 2.85 | 14.00 | -- | -- |
| | Class. acc. (%) | 98.36 | 94.50 | -- | -- |
| SNR | No. of features | 6.65 | -- | 9.45 | 29.00 |
| | Class. acc. (%) | 86.44 | -- | 86.26 | 59.10 |

*ii) Subset Evaluation*: In ACOFS, we evaluate all subsets $S$ using constructive NN in which four kinds of operations are necessarily required for each case, such as, (a) partial training, (b) termination criterion, (c) further training, (d) contribution computation, and (e) adding a hidden neuron. If we analyze and summarize then it can be found that the computation takes O$(k \times \tau_e \times M \times T \times P_t \times W)$ operations for evaluating all $S$ in $T$ iterations. Here, $k$ is the number of constructed $S$ in one iteration. $\tau_e$ and $M$ refer to the number of epochs in one partial training and total number of partial training, whereas $P_t$ and $W$ denote the number of examples in the training set, and the number of weights in the current NN, respectively.

For the remaining steps it can be found that there are some constant and negligible computations are taken place. Thus, finally we get the total computational cost by neglecting and summarizing: O$(k \times r \times q)$+O$(k \times \tau_e \times M \times T \times P_t \times W)$. Since, O$(k \times r \times q)$ << O$(k \times \tau_e \times M \times T \times P_t \times W)$, hence the total computational cost of ACOFS is O$(k \times \tau_e \times M \times T \times P_t \times W)$, which is almost similar to the existing FS algorithm [12]. Thus, it is clear that the incorporation of several techniques in ACOFS does not increase its computational cost.

## 4 Conclusion

In this paper, an efficient FS algorithm ACOFS is presented that is based on the ACO algorithm. The proposed ACOFS utilizes the strategy of real ants for finding shortest path from nest to food sources in FS task. A new modification between the pheromone update rules and the estimation of heuristic information was done in this proposed algorithm in order to enhance the performance of ACO algorithm for FS.

In ACOFS, neither the incorporation of forward selection strategy nor fixed size subset strategy is taken into account for its feature SC process. Apart from these both issues, ACOFS uses a probabilistic scheme for determining the subset size adaptively. Such incorporation ultimately guides to find the best solutions in terms of lower number of salient features and of better classification performances.

Extensive experiments have been carried out in this paper to evaluate how well ACOFS performed on different real-world datasets in comparison with three prominent FS algorithms. In almost all except only for once, ACOFS outperformed the others in terms of the number of selected features and classification performances. The results of the low standard deviations of the classification accuracies exhibit the robustness of this algorithm. Furthermore, the estimated computational complexity of this algorithm reflected that incorporation of several techniques does not increase the computational cost during FS in comparison of other ACO based FS algorithms.

Since the focus of this paper is to present the fundamental ideas and technical details of ACOFS, the detailed comparisons with other algorithms using rigorous statistical methods are left as the future work.

## Acknowledgements

Supported by grants to KM from the Japanese Society for Promotion of Sciences, the Yazaki Memorial Foundation for Science and Technology, and the University of Fukui.

## References

1. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: 17th International Conference on Machine Learning (2000)
2. Gasca, E., Sanchez, J.S., Alonso, R.: Eliminating redundancy and irrelevance using a new MLP-based feature selection method. Pattern Recognition 39, 313–315 (2006)
3. Guan, S., Liu, J., Qi, Y.: An incremental approach to contribution-based feature selection. Journal of Intelligence Systems 13(1) (2004)
4. Abe, S.: Modified backward feature selection by cross validation. In: Proceedings of the European Symposium on Artificial Neural Networks, pp. 163–168 (2005)
5. Yang, J.H., Honavar, V.: Feature subset selection using a genetic algorithm. IEEE intelligent systems 13(2), 44–49 (1998)
6. Muni, D.P., Pal, N.R., Das, J.: Genetic Programming for Simultaneous Feature Selection and Classifier Design. IEEE Trans. on Syst., Man, and Cybern.-Part B: Cybern. 36(1) (2006)
7. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: Optimization by a colony of cooperative agents. IEEE Trans. on Systems, Man, and Cybern. 26(1), 29–41 (1996)
8. Maniezzo, V., Colorni, A.: The ant system applied to the quadratic assignment problem. IEEE transaction on knowledge and data engineering 11(5), 769–778 (1999)
9. Aghdam, M.H., Aghaee, N.G., Basiri, M.E.: Text feature selection using ant colony optimization. Expert systems with applicantions 36, 6843–6853 (2009)

10. Kanan, H.R., Faez, K., Taheri, S.M.: Feature selection using ACO: a new method and comparative study in the application of face recognition system. In: Intl. conf. on data mining, pp. 63–76 (2007)
11. Khushaba, R.N., Alsukker, A., Ani, A.A., Jumaily, A.A.: Enhanced feature selection algorithm using ant colony optimization and fuzzy memberships. In: Intl. conf. on biomedical engineeri., pp. 34–39 (2008)
12. Sivagaminathan, R.K., Ramakrishnan, S.: A hybrid approach for feature subset selection using neural networks and ant colony optimization. Expert systems with applications 33, 49–60 (2007)
13. Ani, A.: Feature subset selection using ant colony optimization. International journal of computational intelligence 2, 53–58 (2005)
14. Ke, L., Feng, Z., Ren, Z.: An efficient ant colony optimization approach to attribute reduction in rough set theory. Pattern Recognition Letters 29, 1351–1357 (2008)
15. Robbins, K.R., Zhang, W., Bertrand, J.K.: The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification. J. of Math. Medi. and Biol., 1–14 (2008)
16. Kwok, T.Y., Yeung, D.Y.: Objective functions for training new hidden units in constructive neural networks. IEEE Trans. Neural Network 8(5), 1131–1148 (1997)
17. Goldberg, D.E.: Genetic Algorithms in search, optimization and machine learning (2004)
18. Mitchell, T.M.: Machine learning. McGraw-Hill, New York (1997)
19. Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
20. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases, Dept. of Information and Computer Sciences, University of California, Irvine (1998)
21. Rumelhart, D.E., McClelland, J.: Parallel distributed processing. MIT Press, Cambridge (1986)

# Stable Training Method for Echo State Networks Running in Closed-Loop Based on Particle Swarm Optimization Algorithm

Qingsong Song, Zuren Feng, and Yonggang Wang

System Engineering Institute, Xi'an Jiaotong University. Xi'an 710049, China
qssong@sei.xjtu.edu.cn

**Abstract.** Echo state network (ESN) is a new paradigm for using recurrent neural networks (RNNs) with a simpler training method, where an RNN is generated randomly and only a readout is trained. ESN method has quickly become popular in modeling for nonlinear dynamic systems. However, the classical training method for ESNs can not ensure the dynamics asymptotic stability any more if the trained ESNs run in a closed-loop self-generative mode. The reason is analyzed at first. We then consider the ESN training problem as an optimization problem with a nonlinear constraint, and take a particle swarm optimization (PSO) algorithm solve it. In the simulation experiments, the ESNs are trained as "figure-eight" trajectory generators. The results show that the proposed PSO-based training method can effectively ensure the dynamics asymptotic stability as well as the precision of generating trajectories of the trained ESNs.

**Keywords:** Echo state networks, Particle swarm optimization, Recurrent neural networks, Asymptotic stability.

## 1 Introduction

As a novel artificial recurrent neural network (RNN) modeling method for dynamical systems, echo state network (ESN) has quickly become popular since being proposed [1]. ESN provides rich dynamics along with a fast and efficient learning algorithm. And besides, ESN overcomes the shortcomings of previous RNN learning algorithms (backpropagation through time, real time recurrent learning, etc.), such as slow or non-convergence, local minimum, and time-consuming [2]. Though ESN is initially proposed for time series predictions and indeed performs remarkably well on this kind of data, it has been generalized to many other application domains [3]. In the literature, the inherent superiorities of ESNs are demonstrated, such as easy training, good generalization capability, and good robustness against noise.

However, in the literature the asymptotic stability of ESNs running with output feedbacks is not given sufficient attention. The asymptotic stability can not be ensured by the spectral radius of the internal connection weight matrixes any more if the trained ESNs run with output feedbacks. There are few contributions studying this problem. The trick—stabilization through wobbling—is the most common method used for

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 253–262, 2009.

achieving the stability [4]; but, the related noise amount is hard to set correctly for specific tasks. In this paper, we propose another training method based on particle swarm optimization (PSO) algorithm for the ESNs running in closed-loop, which considers requirements of the prediction output accuracy as well as the asymptotic stability. Simulation experiment results show that the proposed PSO-based training method is indeed effective.

The remainder of this paper is organized as follows. Section 2 reviews the classical ESN generating and training method and clarifies why the asymptotic stability loses for the trained ESNs running in closed-loop. The proposed PSO-based training method is described in section 3. In section 4, simulation experiments are made to test effectiveness of the proposed method. Finally, discussion and conclusion are drawn in section 5.

## 2 Echo State Networks and Its Asymptotic Stability

### 2.1 Classical Echo State Networks

In general, the classical ESN consists of three parts as illustrated with Fig. 1: the front is input, the mid is a large RNN (it is called dynamical reservoir (DR) [1]), and the end is a readout. It is assumed that the ESN has $L$ input neurons, $N$ reservoir neurons and $M$ readout neurons. At time step $k$, the activations of input neurons are $\mathbf{u}(k)=(u_1(k),u_2(k),...,u_L(k))$, of reservoir neurons are $\mathbf{x}(k)=(x_1(k),x_2(k),...,x_N(k))$, and of readout neurons are $\mathbf{y}(k)=(y_1(k),y_2(k),...,y_M(k))$. Weights for the input connection are in matrix $\mathbf{W}^{in}$, for the reservoir connection in $\mathbf{W}$, for the output connection in $\mathbf{W}^{out}$ and for the connection from the readout output to the reservoir neurons in $\mathbf{W}^{back}$. The corresponding sizes are $N\times L$, $N\times N$, $M\times N$ 和 $N\times M$. Here assuming there are no direct connections from input neuron to readout neuron and no connections between readout neurons.

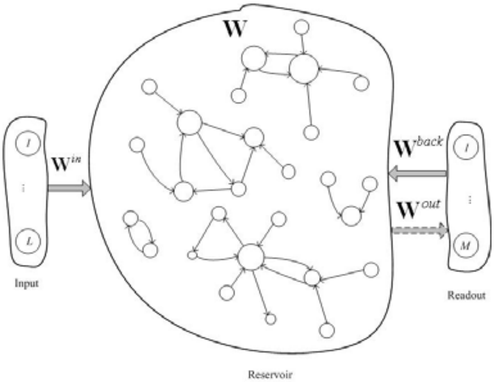


**Fig. 1.** System architecture of ESN

Three phases are included in an implementation process, i.e., the initialization, the training and the exploitation [5]. We assume in this paper that ESNs are trained as some trajectory-generators.

### 1) Initialization

Generate randomly the matrixes ( $\mathbf{W}^{in}$ , $\mathbf{W}$ , $\mathbf{W}^{back}$ ) according to Equation 1~3, where the constants $a$ , $b$ , $c$ and $d$ are the scales and shifts for $\mathbf{W}^{in}$ and $\mathbf{W}^{back}$ ; $\mathrm{rand}$ and $\mathrm{sprand}$ are Matlab functions, the former generates uniformly distributed random numbers, the latter generates a sparse uniformly distributed random matrix; $|\lambda_{\max}|$ is the spectral radius of $\mathbf{W}_0$ ; $D$ and $\rho_{\mathrm{W}}$ respectively stand for the connectivity density and the spectral radius of $\mathbf{W}$ . After having been generated in such manner, these three matrixes are fixed all along in the implementation process.

$$\mathbf{W}^{in} = a(\mathrm{rand}(N, L) - b) \tag{1}$$

$$\mathbf{W}^{back} = c(\mathrm{rand}(N, M) - d) \tag{2}$$

$$\begin{aligned} &\mathbf{W}_0 = \mathrm{sprand}(N, N, D) \\ &\mathbf{W}_0 = \mathbf{W}_0 / |\lambda_{\max}| \\ &\mathbf{W} = \rho_{\mathrm{W}} \mathbf{W}_0 \end{aligned} \tag{3}$$

### 2) Training

From the desired trajectory we get a training sequence $\mathbf{y}_d(k)$ , $k = 1, 2, ..., T_t$ . The reservoir state is initialized with $\mathbf{x}(0) = \mathbf{0}$ . The generating ESN ( $\mathbf{W}^{in}$ , $\mathbf{W}$ , $\mathbf{W}^{back}$ ) is driven by teacher-forcing the training sequence as Equation 4, where $f$ is the reservoir neurons' activation function (hyperbolic tangent function in this paper); $\mathbf{u}(k)$ is often a constant bias input; $\mathbf{v}(k)$ is a noise term sampled from a uniform distribution over [ $-e$ , $e$ ].

$$\mathbf{x}(k+1) = f(\mathbf{W}^{in}\mathbf{u}(k+1) + \mathbf{W}\mathbf{x}(k) + \mathbf{W}^{back}\mathbf{y}_d(k) + \mathbf{v}(k+1)) \tag{4}$$

From time step $T_0 + 1$ to $T_t$ , $T_0 < T_t$ , the reservoir states $\mathbf{x}(k)$ s are collected row-wise in a state collecting matrix $\mathbf{M}$ . At the same time, the sigmoid-inverted training data $(f^o)^{-1}(\mathbf{y}_d(k))$ s are collected row-wise in another matrix $\mathbf{T}$ , here $f^o$ is the readout neurons' activation function.

The output weight matrix $\mathbf{W}^{out}$ is computed simply with Equation 5, where $[\cdot]^{-1}$ is the inverse matrix of $[\cdot]$ , $[\cdot]'$ is the transpose matrix of $[\cdot]$ .

$$\mathbf{W}^{out} = (\mathbf{M}^{-1}\mathbf{T})' \tag{5}$$

Now, we obtain the trained ESN ( $\mathbf{W}^{in}$ , $\mathbf{W}$ , $\mathbf{W}^{out}$ , $\mathbf{W}^{back}$ ).

### 3) Exploitation

The trained ESN is driven by an exploitation sequence $\mathbf{y}(k)$ , $k = 1, 2, ..., T_e$ . From $k = 1$ to $k = T_0$ , the reservoir state is updated by Equation 4. In this period, the outputs are no significant and discarded. However, from $T_0+1$ to $T_e$, the reservoir state $\mathbf{x}(k)$ and prediction output $\hat{\mathbf{y}}(k)$ are updated as follows.

$$\mathbf{x}(k+1)=f(\mathbf{W}^{in}\mathbf{u}(k+1)+\mathbf{W}\mathbf{x}(k)+\mathbf{W}^{back}\hat{\mathbf{y}}(k)) \tag{6}$$

$$\hat{\mathbf{y}}(k+1)=f^{o}(\mathbf{W}^{out}\mathbf{x}(k+1)) \tag{7}$$

### 2.2 Problem on the Asymptotic Stability

Echo state property is expected to prevent ESNs from having a chaotic, unbounded behavior [6]. In the initialization phase, in order to ensure echo state property the spectral radius $\rho_{\mathrm{W}}$ is assigned to be smaller than unit. It is correct for the ESNs running in opened-loop; however, for the ESNs running in closed-loop, it is wrong.

Assuming $f^{o}$ is identity function and $\mathbf{u}(k)\equiv\mathbf{0}$. By substituting $\mathbf{W}^{out}\mathbf{x}(k)$ for $\hat{\mathbf{y}}(k)$ in Equation 6, we obtain Equation 8. It can be see that during the exploitation, the trained ESN is an autonomous difference dynamical system with an initial value $\mathbf{x}(T_0+1)$.

$$\mathbf{x}(k+1)=f((\mathbf{W}+\mathbf{W}^{back}\mathbf{W}^{out})\mathbf{x}(k)) \tag{8}$$

$$\tilde{\mathbf{W}}=(\mathbf{W}+\mathbf{W}^{back}\mathbf{W}^{out}) \tag{9}$$

Hyperbolic tangent function satisfies the Lipschitz condition [7], so the asymptotic stability of the trained ESN is only determined by the property of the matrix $\tilde{\mathbf{W}}$ (Equation 9). According to the idea of the proving method for the proposition 3 in [4], it can be proved that it is the spectral radius of $\tilde{\mathbf{W}}$ (we call it effective spectral radius (ESR), and note it as $\rho(\tilde{\mathbf{W}})$) rather than $\mathbf{W}$ which determines the trained ESN's stability.

## 3 Training Method Based on Particle Swarm Optimization Algorithm

Based on Equation 9, it can be said that $\mathbf{W}^{out}$ not only determines the prediction accuracy, but also has effect on the asymptotic stability of the trained ESNs. The classical training method as Equation 5 takes the prediction accuracy into account, neglects the requirement for the asymptotic stability, however. We consider it as an important factor which induces the unstable results in [8]. Here by considering both the precision and the stability in a comprehensive way, we translate the training problem into a continuous optimization problem with a nonlinear constraint, which is illustrated with Equation 10.

$$\begin{aligned}\min \quad & \sum_{k=T_0+1}^{T_t}(\mathbf{y}_d(k)-\mathbf{W}^{out}\mathbf{x}(k))^2/(T_t-T_0)\\ s.t. \quad & \rho(\widetilde{\mathbf{W}})<1\end{aligned} \tag{10}$$

We find that gradient-based deterministic methods can hardly solve Equation 10 due to the highly nonlinear constraint. PSO is a stochastic optimization algorithm modeled on swarm intelligence, and it has been demonstrated that in many research and application areas, PSO algorithm can find a better solution in a search space in a faster, cheaper way compared with other methods [9]. We choose PSO algorithm to solve Equation 10.

Since the reservoir of ESN does not have to be trained for a particular task, it supports parallel computing: multiple readout neurons can be independently trained to perform different tasks in parallel [10]. If $M$ >1, the readout neurons are trained one by one by the PSO-based method in this paper.

Firstly, the objective function and the constraint in Equation 10 are integrated into a single evaluation function $\Phi(\mathbf{W}^{out})$.

$$\Phi(\mathbf{W}^{out}) = g(\mathbf{W}^{out}) + r \cdot \theta \cdot h(\mathbf{W}^{out}) \tag{11}$$

$$g(\mathbf{W}^{out}) = \sum_{k=T_0+1}^{T_t} (\mathbf{y}_d(k) - \mathbf{W}^{out}\mathbf{x}(k))^2 / (T_t - T_0) \tag{12}$$

$$h(\mathbf{W}^{out}) = \max[0, (\rho(\widetilde{\mathbf{W}}) - 1)] \tag{13}$$

where $g(\mathbf{W}^{out})$ measures the mean squared training error (MSE); $h(\mathbf{W}^{out})$ characters violation amount of the restraint; $r$ and $\theta$ are dynamic parameters of the evaluation function.

And subsequently, PSO algorithm is called to solve Equation 11. The swarm has $N_s$ particles; each particle has a position $\mathbf{p}$ and a velocity $\mathbf{V}$ in the $N$ dimensional space. Here, $\mathbf{p}$ corresponds to $\mathbf{W}^{out}$ , i.e., each value of $\mathbf{p}$ denotes a possible solution of Equation 10. At each evolution step ( $step$ ), $\mathbf{p}$ and $\mathbf{V}$ are updated by Equation 14 and 15.

$$\mathbf{V}(step+1) = \chi(\mathbf{V}(step) + c_1 \cdot \text{rand} \cdot (\mathbf{p}_{best}(step) - \mathbf{p}(step)) + c_2 \cdot \text{rand} \cdot (\mathbf{g}_{best}(step) - \mathbf{p}(step))) \tag{14}$$

$$\mathbf{p}(step+1) = \mathbf{p}(step) + \mathbf{V}(step+1) \tag{15}$$

where $\chi$ is constriction factor, $c_1$ and $c_2$ are acceleration coefficients; $\mathbf{p}_{best}$ and $\mathbf{g}_{best}$ are the particle's and the whole swarm's best position at current evolution step, respectively. Here, to ensure the convergence, we use the PSO algorithm with a constriction factor, $\chi$ = 0.729, $c_1$ = $c_2$ = 1.496 [11,12].

The proposed PSO-based training method is implemented as follows.

s1. Initialization. Once $\mathbf{M}$ and $\mathbf{T}$ have been obtained, set $N_s$ , $r$ , $\theta$ and $step_m$ (the maximum of evolution step). Randomly place all particles in the $N$ dimensional space, and initialize their velocities. Each particle's current position is recorded as its best position $\mathbf{p}_{best}(1)$ . Among all $N_s$ $\mathbf{p}_{best}(1)$ s, the one having the minimum $\Phi(\mathbf{W}^{out})$ is recorded as $\mathbf{g}_{best}(1)$ .

s2. Update $\mathbf{p}$ and $\mathbf{v}$. Update each particle according with Equation 14 and 15, obtain a new swarm.

s3. Evaluation. Substitute new $\mathbf{p}$ for $\mathbf{W}^{out}$ in Equation 11; get new evaluation for each particle.

s4. Update $\mathbf{p}_{best}$ and $\mathbf{g}_{best}$ . For each particle, compare its $\Phi(\mathbf{p}(step))$ with $\Phi(\mathbf{p}_{best}(step-1))$ . If $\Phi(\mathbf{p}(step)) < \Phi(\mathbf{p}_{best}(step-1))$ , then $\mathbf{p}(step) \to \mathbf{p}_{best}$ . Note $\Phi_{\min}$ as the minimum of all $\Phi(\mathbf{p}(step))$ s. Compare $\Phi_{\min}$ and $\Phi(\mathbf{g}_{best})$ . If $\Phi_{\min} < \Phi(\mathbf{g}_{best})$ , then $\mathbf{p}(step)|_{\Phi(\mathbf{p}(step))=\Phi_{\min}} \to \mathbf{g}_{best}$ .

s5. Repeat s2~s4 until $step = step_m$ . Now, the resulting $\mathbf{g}_{best}$ is just the trained result of $\mathbf{W}^{out}$ .

## 4 Simulation Experiments

The "figure eight" generation task has been considered as a benchmark problem for ESNs to test whether the trained ESNs are stable or not. A "figure eight" can be simply interpreted as the superposition of a sine (for the x direction) and a cosine of half the sine's frequency (for the y direction) [8], as shown in Fig. 2.

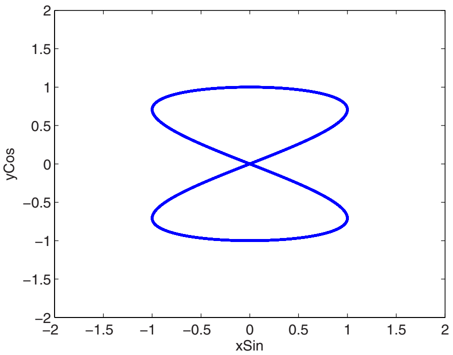


**Fig. 2.** Schematic diagram of the "figure eight"

Two ESN groups are independently tested on the same "figure eight" task. Each group consists of 20 independently generated ESNs ( $N_e$ =20). One group is trained by the classical method described in section 2; the other is trained by the proposed PSO-based method.

The training sequence and exploitation sequences are processed as the manner being used in [8]: the figure-eight trajectory is discrete and one full ride along the eight consists of 200 sample points.

**Table 1.** Parameter assignments in the simulation experiments

| $a$ | $b$ | $c$ | $d$ | $\rho_{\mathrm{W}}$ | $D$ | $L$ | $N$ | $M$ | $e$ | $T_0$ | $T_t$ | $T_e$ | $N_s$ | $step_m$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 0.5 | 2 | 0.5 | 0.8 | 0.8 | 0 | 20 | 2 | 1e-6 | 1,000 | 3,000 | 100,000 | 30 | 500 |

Assignments to the parameters involved in the simulation experiments are listed in Table 1. The dynamic parameters of the evaluation function are set as Equation 16 and 17.

$$r = step^{3/2} \tag{16}$$

$$\theta = \begin{cases} 50, & \text{if } h(\mathbf{W}^{out}) > 1; \\ 20, & \text{if } 0.5 < h(\mathbf{W}^{out}) \le 1; \\ 2, & \text{if } 0.2 < h(\mathbf{W}^{out}) \le 0.5; \\ 1, & \text{if } h(\mathbf{W}^{out}) \le 0.2; \end{cases} \tag{17}$$

The performances are evaluated with two indexes. One is the mean squared error during the exploitation (exploitation MSE), and the other is the successful training ratio, which is defined as Equation 18.

$$\begin{gathered} s_r = \sum_{i=1}^{N_e} \delta(1-\rho(\widetilde{\mathbf{W}}))/N_e \\ \delta(x_v) = \begin{cases} 1 & x_v > 0 \\ 0 & x_v \le 0 \end{cases} \end{gathered} \tag{18}$$

where $N_e$ is the number of ESNs which are randomly independently generated.

Fig. 3 and Fig. 4 respectively give the evolution curves of $g(\mathbf{W}^{out})$ and $\rho(\widetilde{\mathbf{W}})$. It can be seen that both $g(\mathbf{W}^{out})$ and $\rho(\widetilde{\mathbf{W}})$ have already converged about at $step$ =300. For the x direction data, the corresponding ESR converges at 0.99997, as being shown with the red dashed curve (with legend "m = 2"); while for the y direction data, the ESR converges at 0.99994, as being shown with the blue solid curve (with legend "m = 1"). So echo state property is ensured, the trained ESNs have the asymptotic stability.

Fig. 5 demonstrates the generated trajectory from an ESN trained by the classical method (red dashed curve). The blue solid curve denotes a true trajectory of the "figure-eight". It can be seen that the ESN initially generates a crude "figure-eight"; however, the trajectory in the x direction loses its stability quickly: the x-coordinate data is far away from [-1, 1], and the corresponding ESR value is about 11.2 for the output neuron generating the x direction data.

Fig. 6 demonstrates a predicting output trajectory of an ESN trained by the proposed PSO-based method. The output weight values correspond to the position of the best particle at the 100th evolvement step of the PSO algorithm. The trained ESN is stable: the ESR value is about 0.99997 for the output neuron generating the x direction data, about 0.99949 for another output neuron generating the y direction data. But the prediction precisions are insufficient: the exploitation MSE value is about 0.16 for the x direction data, about 0.28 for the y direction data.

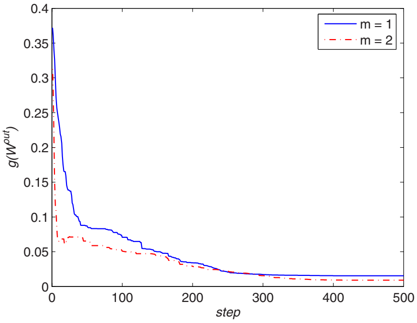


**Fig. 3.** Evolution curves of the training MSE. “m=1” denotes one readout neuron, of which the corresponding training MSE evolution curve is denoted as the blue solid curve. “m=2” denotes another readout neuron, of which the corresponding training MSE evolution curve is denoted as the red dashed curve.

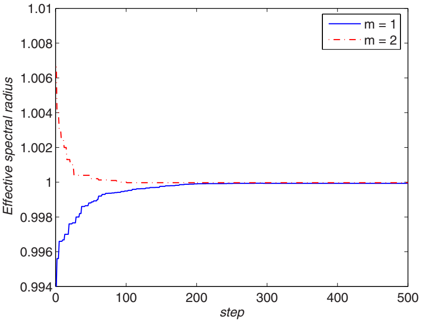


**Fig. 4.** Evolution curves of the effective spectral radius. “m=1” denotes one readout neuron, of which the corresponding ESR evolution curve is denoted as the blue solid curve, “m=2” denotes another readout neuron, of which the corresponding ESR evolution curve is denoted as the red dashed curve.

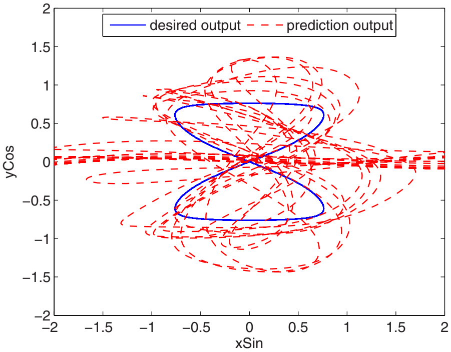


**Fig. 5.** Prediction output curve of an ESN trained by the classical method

Fig. 7 also demonstrates a predicting output trajectory of another ESN trained by the same PSO-based method; however, the output weight values correspond to the position of the best particle at the 500th evolvement step of the PSO algorithm. The trained ESN is also stable: the corresponding ESR values are about 0.99997 and 0.99994 respectively. But in this case, the prediction precisions are sufficient: the corresponding MSE values are about 0.08 and 0.12 respectively.

The noise-robustness of the ESNs trained by the proposed PSO-based method is also verified. While the ESNs run in self-generative mode, their predicting outputs are perturbed with Gaussian noise (the mean: 0, the variance: 0.05) from time steps 1001~1200, i.e., just one full ride along the figure-eight. After that period, the noise is

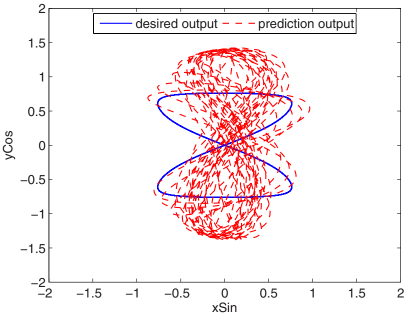


**Fig. 6.** Prediction output curve of an ESN trained by the proposed PSO-based method. The particle swarm is evolved over 100 steps ( *step* =100).

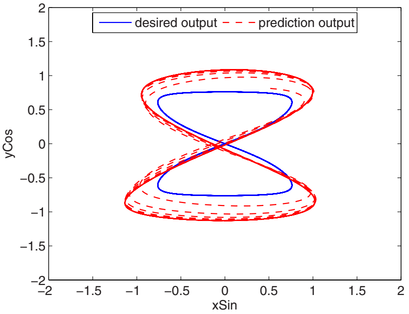


**Fig. 7.** Prediction output curve of an ESN trained by the proposed PSO-based method. The particle swarm is evolved over 500 steps ( *step* =500).

dispelled. It is demonstrated from the experiment results that about 900 time steps (four and a half full rides along the figure-eight) after having been perturbed, the ESNs have already converged back to the “figure-eight” attractor. The PSO trained ESNs have good noise-robustness.

The simulation experiment results are summarized in Table 2, they are the averages over the results of all 20 randomly independently generated ESNs ( $N_e$ = 20). The classical training method described in section 2 can not ensure the ESNs stability: the exploitation MSE values are always far larger than 2; the ESR values are far larger than unit; the obtain $s_r$ equals zero for the x direction data, equals 0.4 for the y direction data. As a comparison, the proposed PSO-based training method can effectively assure the stability of the trained ESNs as well as the precision of the generated “figure-eight” trajectories: the obtained ESR values are always smaller than unit (unexpectedly, the ESRs nearly approach unit); and the $s_r$ values equal 1 as the PSO algorithm has evolved over 500 steps.

**Table 2.** Contrast of the result between the classical and the PSO-based training method

| / | MSE (exploitation) | | ESR | | $s_r$ | |
|---|---|---|---|---|---|---|
| | m = 1 | m = 2 | m = 1 | m = 2 | m = 1 | m = 2 |
| The classical method | $\gg 2$ | $\gg 2$ | 6.23186 | 11.8225 | 0.4 | 0 |
| PSO-based method ( *step* =100) | 0.28 | 0.16 | 0.99949 | 0.99997 | 1 | 0.9 |
| PSO-based method ( *step* =500) | 0.12 | 0.08 | 0.99994 | 0.99997 | 1 | 1 |

## 5 Conclusion and Discussion

We have reviewed the classical ESN paradigm, and pointed out that the asymptotic stability of the trained ESNs running in closed-loop can not be ensured any more by the parameter-setting that the spectral radius of reservoir connection weight matrix is smaller than unit. And then we translate the output weight training problem into a nonlinearly constrained optimization problem, which is solved by the PSO algorithm. The simulation experiment results show that the proposed PSO-based training method can effectively ensure the stability as well as the precision of the generating trajectories of the trained ESNs.

## Acknowledgements

This work is supported in part by the National Natural Science Foundation of China under Grant No. 60875043 and the State Key Development Program for Basic Research of China under Grant No. 2007CB311006.

## References

1. Jaeger, H., Haass, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science 304, 78–80 (2004)
2. Haykin, S.: Neural networks – a comprehensive foundation. Prentice-Hall, NJ (1999)
3. Lukoševičius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Computer Science Review (in press) doi:10.1016/j.cosrev.2009.03.005
4. Jaeger, H.: The echo state approach to analyzing and training recurrent neural networks. GMD Report 148, German National Research Center for Information Technology (2001)
5. Jaeger, H.: Tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the "echo state network" approach. Tech. Rep. No. 159, Bremen: German National Research Center for Information Technology (2002)
6. Jaeger, H.: Echo state network. Scholarpedia 2, 2330 (2007)
7. Oden, J.T.: Applied Functional Analysis. Prentice-Hall, NJ (1979)
8. Jaeger, H., Lukoševičius, M., Popovici, D., Siewert, U.: Optimization and applications of echo state networks with leaky-integrator neurons. Neural Networks 20, 335–352 (2007)
9. Clerc, M.: Particle Swarm Optimization. Wiley-ISTE (2006)
10. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation 14, 2531–2560 (2002)
11. Eberhart, R.C., Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the 2000 Congress on Evolutionary Computation, vol. 1, pp. 84–88 (2000)
12. Clerc, M.: The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. In: Proc. 1999 ICEC, Washington, DC, pp. 1951–1957 (1999)

# A Concept Generation Method Based on Mutual Information Quantity among Multiple Self-organizing Maps

Kunio Kitahara and Akira Hirose

Department of Electronic Engineering, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
kitahara@eis.t.u-tokyo.ac.jp, ahirose@ee.t.u-tokyo.ac.jp
http://www.eis.t.u-tokyo.ac.jp/

**Abstract.** We propose a concept generation method based on mutual information smong multiple self-organizing maps. There have been many reports on concept generation in various fields such as linguistics and robotics. In the context of language acquisition, however, they mainly deal with letters, i.e. symbols, in general. Since both the concept and language acquisitions progress in parallel, we notice the importance to investigate concept generation without symbols existing *a priori*. In this paper, we propose a non-symbol-based method that pays attention to multimodal mutual information.

## 1 Introduction

Recent progress in humanoid robotics has brought about various human-like acts and behaviors such as bipedal walking. Humanoid robots in the future will be expected to act and behave adaptively just like human beings, or more flexibly in specific fields. Learning and self-organization will play an important role more and more in this context.

When we focus on the learning of communications, language acquisition is the central problem. From the viewpoint of cognitive science, we need to elucidate the generation of concept. In this paper, we investigate a method to generate concept in multimode information by paying attention to mutual information quantity among self-organizing maps (SOMs).

There has been two streams in human-machine communications. ELIZA constructed by Weizenbaum in 1966 [1] generates sentences that prompt human beings to make a reply according to a set of simple rules without limitation on topics. It is called *chatterbot*. SHRDLU desined by Winograd in 1972 [2] behaves as it is told, e.g., "lift the block." However, the robots just obey built-in rules. They can recognize only what they already know. People soon get tired of the *conversation*. We expect near-future robots to learn languages. An example is the infanoid [3], 480mm in the seated height, capable of repeating what it hears in a parrot fashion. Eyes on the head provide visual signal so that it can detect

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 263–272, 2009.

and track target faces and toys. Ears catch speech so that it extracts prosodic information as well as phonological information, which are fed to a voice synthesizer to repeat the speech. Mechanisms in the human brain are still unclear, but they are expected to contribute future humanoid robots that can learn and self-organize.

There are various ideas on what is concept, and on how to define concept generation. We may be able to show rough three themes in this field. They are, namely, how to acquire information in the environment, how to store the information, and how to find relationship among a number of sets of the information.

Regarding the acquisition, it is important to elucidate how to construct a set of features. For example, Iwahashi et al. [4] investigated the acquisition of abstract concept concerning spatial relationship. The storage is studied often in conjunction with the discovery of relationship. In particular in concept generation, the way of store is considered together with symbolic grounding. Accumulated information often biases acquisition and store of new information. Recent considerations on the Booba-kiki test [5] and the column structure in the cerebrum suggest possible emergence of creativity by being based on a mapping sensitive to similarity in multimodal information. In this context, the relationship among multiple sets of information will also serve a critical role.

The symbolic grounding is usually a process to tag each image with a text. However, we have to prepare a set of texts a priori. Texts are symbols having no natural metric among them. Then we cannot tag an image with unknown words or new languages. Roy & Pentland [6] and Roy & Mukherjee [7] used mutual information to relate respective images to sound data though they do not refer to concept generation. Sound data, unlike texts, are non-symbolic information. The relating process is free from supervisor, and flexible enough to accept unknown words or languages. The method should also be useful in concept generation. This paper proposes a method to realize clustering of similar data based on mutual information, which can work as a concept generation process.

## 2 Mapping in SOMs Modulated by Mutual Information

### 2.1 Basic Idea

Figure 1 is the flow chart showing the concept generation process based on mutual information among multiple SOMs. Each SOM is a basically conventional one [8] [9] that maps input information into low dimensional information space.

Conventionally, a typical SOM dynamics is expressed as the weight updating process in terms of the weights of neurons $\boldsymbol{w}_{\boldsymbol{k}}(t)$, at position $\boldsymbol{k}$ in the SOM space at time $t$ (or learning iteration number $t$), and a set of signal input vector $\boldsymbol{x}_i$ fed to the SOM as

$$\boldsymbol{w}_{\boldsymbol{k}}(t+1) = \boldsymbol{w}_{\boldsymbol{k}}(t) + \Theta(\boldsymbol{k}, \boldsymbol{k}_c; t)\ \alpha(t)\ \left(\boldsymbol{x}_i - \boldsymbol{w}_{\boldsymbol{k}}(t)\right) \tag{1}$$

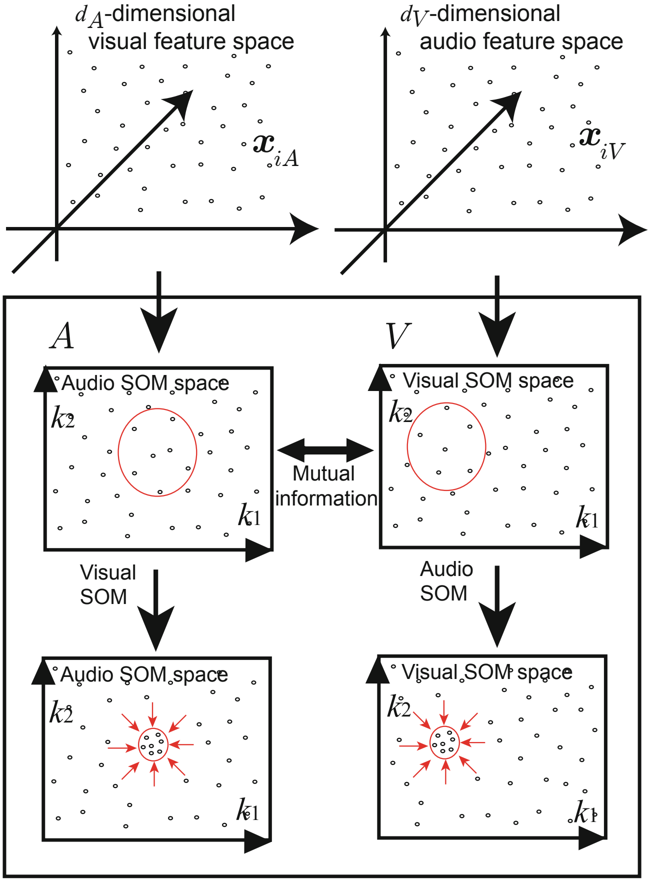


**Fig. 1.** Total processing flow for the concept generation based on mutual information among multiple SOMs

where

$$\boldsymbol{k}_c = \arg\min_{\boldsymbol{k}} \left| \boldsymbol{x}(t) - \boldsymbol{w}_{\boldsymbol{k}}(t) \right| \qquad \text{(winner neuron position)} \qquad (2)$$

$$\alpha(t) \equiv \alpha(0) \left( 1 - \frac{t}{T} \right) \qquad \text{(learning coefficient)} \qquad (3)$$

$$\Theta(\boldsymbol{k}, \boldsymbol{k}_c; t) = \exp \left( - \frac{|\boldsymbol{k} - \boldsymbol{k}_c|^2}{2\delta^2(t)} \right) \qquad \text{(neighborhood function)} \qquad (4)$$

$$\delta(t) \equiv \delta(0) \left( 1 - \frac{t}{T} \right) \qquad \text{(inverse of sharpness of } \Theta) \qquad (5)$$

where $T$ is the maximum time (maximum number of iteration) in the self-organization.

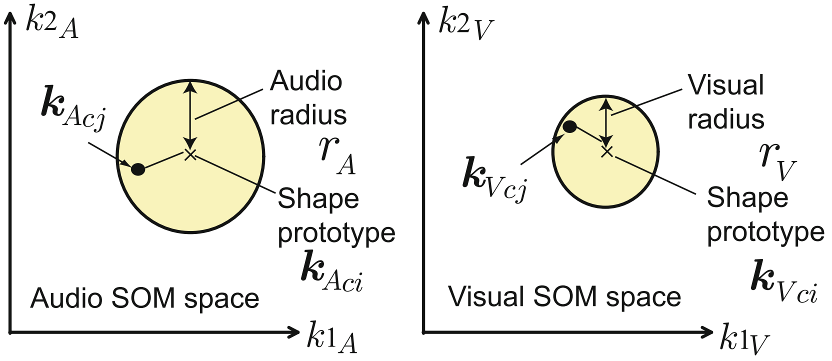


**Fig. 2.** Determination of the region in which we calculate the mutual information to maximize the quantity

However, in our proposal, the dynamics is modulated in accordance with mutual information among input signals to a set of SOMs. That is, a SOM finds correspondence of input information with another input signal fed to another SOM by paying attention to mutual information quantity. Each SOM changes the stretch of its neighborhood according to the mutual information so that a set of corresponding data makes a cluster through self-organization. For example, let's assume visual and audio data streams existing simultaneously. By referring to the mutual information between the visual and audio data, we can segment the visual stream commutatively with the audio data, and vice versa.

### 2.2 SOM Dynamics Modulated by Mutual Information

We propose a set of SOMs, of which dynamics is modulated by mutual information quantity to realize a mapping that takes the relationship among multiple-mode data sets into account. We consider only two modes of data for simplicity here, e.g., visual and audio data streams. We then prepare two SOMs. Each SOM self-organizes by referring to the mutual information and changing the neighborhood function used in the self-organization. We calculate the mutual information between the visual and audio SOM neurons $I(A;V)$ as

$$I(A;V) = \sum_{d_A=0,1} \sum_{d_V=0,1} p(d_A, d_V) \ \log \frac{p(d_A, d_V)}{p(d_A)p(d_V)} \tag{6}$$

$$p(d_A) = \frac{n(d_A)}{N} \tag{7}$$

$$p(d_V) = \frac{n(d_V)}{N} \tag{8}$$

$$p(d_A, d_V) = \frac{n(d_A, d_V)}{N^2} \tag{9}$$

$$d_A = \begin{cases} 1, & \text{if} \quad \left| \boldsymbol{k}_{Aci} - \boldsymbol{k}_{Acj} \right| \leq r_A \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

$$d_V = \begin{cases} 1, & \text{if} \quad \left| \boldsymbol{k}_{Vci} - \boldsymbol{k}_{Vcj} \right| \leq r_V \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where $n(d_A)$ and $n(d_V)$ (or $n(d_{A/V})$ in short) denote the number of the data that gives $d_{A/V}$, $N$ is the total number of the data in the set ( i.e., $\boldsymbol{x}_1 = [\boldsymbol{x}_{A1}, \boldsymbol{x}_{V1}], \boldsymbol{x}_2 = [\boldsymbol{x}_{A2}, \boldsymbol{x}_{V2}], \cdots, \boldsymbol{x}_N = [\boldsymbol{x}_{AN}, \boldsymbol{x}_{VN}]$ ), $\left| \boldsymbol{k}_{A/Vcj} - \boldsymbol{k}_{A/Vci} \right|$ is the distance between the winners for data $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the audio / visual SOM space, and $r_{A/V}$ denotes radius of a variable circle in the SOM space that gives a boundary of the category represented by $\boldsymbol{k}_{A/Vci}$ as shown in Fig.2. We vary the values $r_A$ and $r_V$ to find a pair of $r_A$ and $r_V$ that maximize the mutual information $I_{max}$. Then we modify the neighborhood function in accordance with the $I_{max}$ value as follows.

If the mutual information $I_{max}$ is larger than a threshold $H$, we employ a modified $\delta(t)$ in (5), namely $\tilde{\delta}(t)$ expressed as

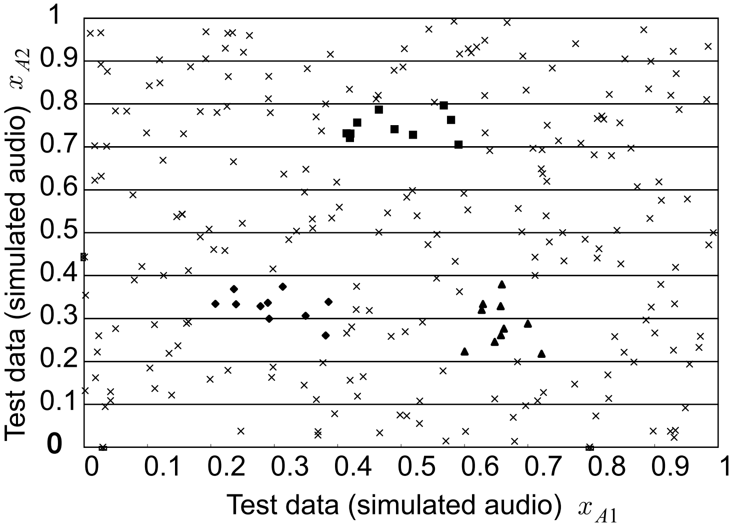
$$\tilde{\delta}(t) = \begin{cases} \delta(0) \ \left(1 - \frac{t}{T}\right)^2 & \text{for} \quad \boldsymbol{k}_{A/Vcj} \subseteq \text{circle of } \boldsymbol{k}_{A/Vci} \\ \delta(0) \ \left(1 - \frac{t}{T}\right) & \text{otherwise} \end{cases} \tag{12}$$

That is, in the upper case in (12), the neighborhood is sharper. In other words, if a winner position $\boldsymbol{k}_{A/Vcj}$ for a data $\boldsymbol{x}_j$ falls within the circle, we employ a smaller $\tilde{\delta}(t)$ in the neighborhood function $\Theta(\boldsymbol{k}, \boldsymbol{k}_c, t)$ when the maximum mutual information $I_{max}$ exceeds the threshold $H$, which indicates a strong correlation between the audio and video data $\boldsymbol{x}_A$ and $\boldsymbol{x}_V$. Though an locally adaptive threshold may improve the dynamics, we employ a fixed $H$ for simplicity in the experiment.

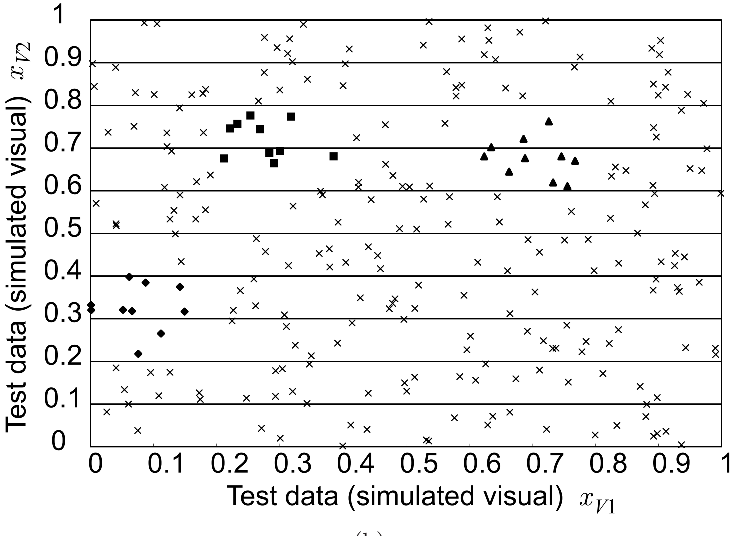
### 2.3 Experiment with Simulation Data

Figure 3(a) and (b) present a set of simulation data in the feature space prepared for a two-modal experiment. We have 250 data points each. The data are numbered sequentially. Data of number 1–10 (circles), 11–20 (squares), and 21–30 (triangles) are distributed within certain regions respectively in each space, while other data points (crosses) are scattered at random in the rest region. Note that the total distribution is uniform so that we cannot observe any clustering of firing neurons when we apply ordinary self-organization dynamics to the respective SOMs separately.

We adopt our proposed method mentioned above. We prepare two SOM spaces. The size of each SOM is 100×100. Parameters in self-organization are $\alpha(0) = 1$(update coefficient of reference vectors in the SOM), $\delta(0) = 30, H = 0.2$, and the number of iterations is 200. Figure 4 shows the result in the SOM

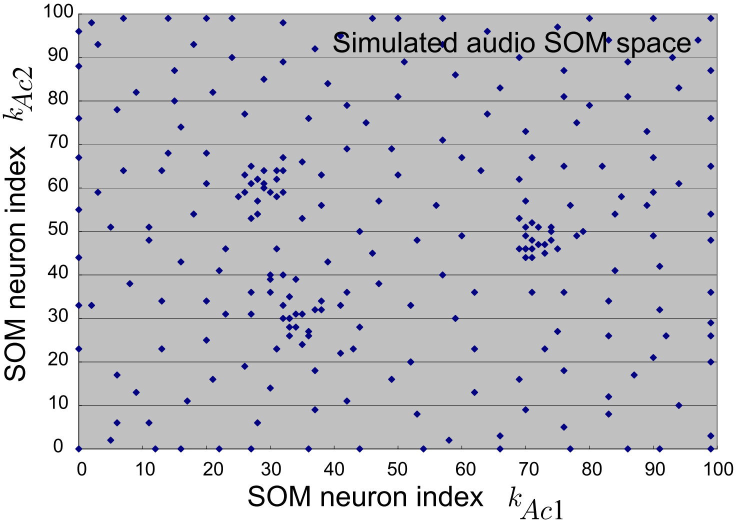


(a)

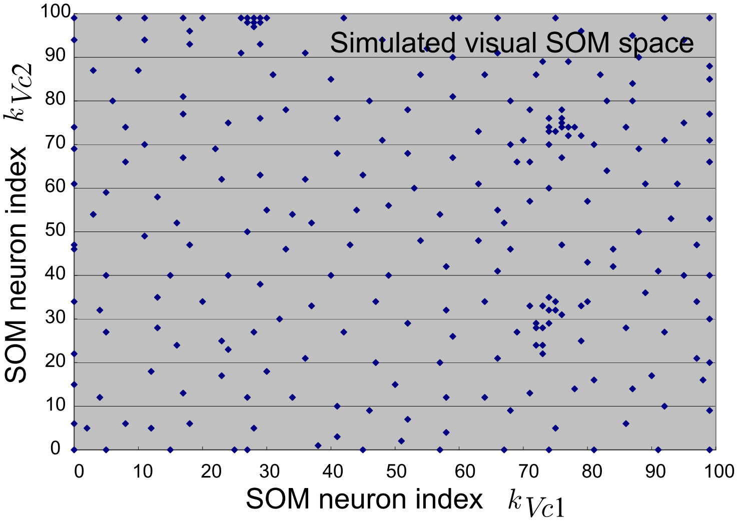


(b)

**Fig. 3.** Simulation test data in feature space in two modes, e.g., (a)audio $\boldsymbol{x}_A$ and (b)audio $\boldsymbol{x}_V$ data. The 250 data points, in total in each feature space, are sequentially numbered and located in such a way that, in visual feature space, data number 1–10 are in $0.2 < x_{A1} < 0.4$ and $0.2 < x_{A2} < 0.4$, while data number 11–20 are in $0.4 < x_{A1} < 0.6$ and $0.7 < x_{A2} < 0.8$, data number 21–30 are in $0.6 < x_{A1} < 0.8$ and $0.2 < x_{A2} < 0.4$, and other data are distributed at random in the rest region. Simultaneously the audio data number 1–10 are in $0 < x_{V1} < 0.2$ and $0.2 < x_{V2} < 0.4$, while data number 11–20 are in $0.2 < x_{V1} < 0.4$ and $0.6 < x_{V2} < 0.8$, data number 21–30 are in $0.6 < x_{V1} < 0.8$ and $0.6 < x_{V2} < 0.8$, and others are distributed at random in the rest region.



(a)



(b)

**Fig. 4.** Distributions of the SOM neurons that fire corresponding to the input data shown in Fig. 3(a) and (b) after the proposed self-organization. There are three clusters showing the three related sets of feature data.

space where the dots indicate firing neurons when the signals $\boldsymbol{x}_A$ and $\boldsymbol{x}_V$ are input. We find three clusters, each of which clusters corresponds to a set of signals of number 1–10, 11–20, or 21–30. We can clearly observe that the self-organization dynamics modified by the mutual information quantity is effective in the clustering, i.e., concept generation.

## 3 Experiment for Real Visual and Audio Data

We examined the effectiveness of our proposal for real-world data clustering. Figure 5 illustrates the process including preprocessing. We deal with each input data as follows. Regarding visual data, we detect edges, and obtain high-dimensional local autocorrelation (HLAC) [10], resulting in 35-dimensional feature vector. Regarding audio data, we first calculate the Mel-frequency cepstrum coefficient (MFCC). Since the dimension depends on the recording time, we adopt the Karhunen-Loeve (KL) expansion to reduce the dimension to 26. We use these two-modal features.

The visual data was prepared from the Amsterdam Library of Object Images (ALOI). Figure 6 shows the data, where a duck is captured around. We used 20 images out of them. The audio data is a set of recorded pronunciation of duck (”ahiru” in Japanese) by a single speaker. The voice was recorded 20 times to generate 20 data with a sampling frequency of 16kHz at 16bit linear predictive coding (LPC). We used the HCopy in the HTKtoolkit [11] for the MFCC with parameters of frame interval of 10ms, frame width of 25ms with the Hamming window, and pre-emphasis index of 0.95. We have one set of correlated data for the duck, and numbered them as 1–20. We also prepared 80 random visual and audio data in addition, and obtained 100 data totally.

We have two SOMs, namely visual and audio SOMs. The size of each SOM is 100×100. Self-organization parameters are $\alpha(0) = 1, \delta(0) = 30, H = 0.2,$

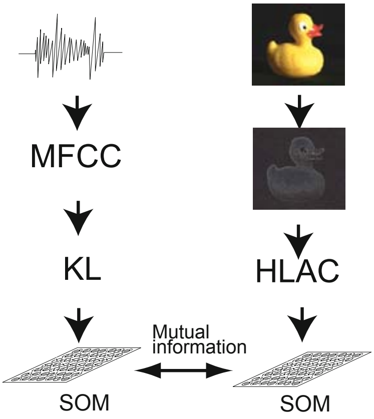
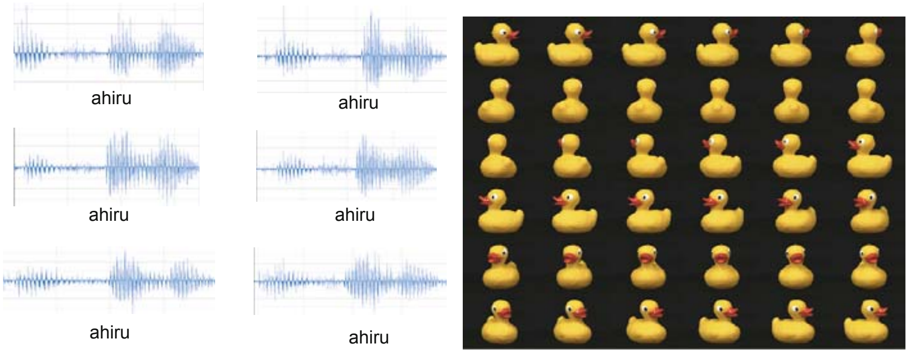
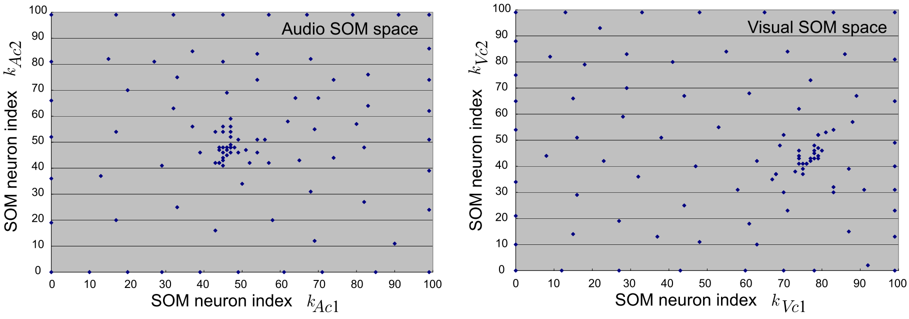


**Fig. 5.** Procedure of the experiment using real audio and visual data



(a) (b)

**Fig. 6.** Data examples: (a)audio waveforms ”ahiru” (duck in Japanese) recorded 20 times for a single speaker and (b)visual images ”duck” captured around



(a) (b)

**Fig. 7.** Distributions of (a)audio- and (b)visual-SOM neurons that fire corresponding to the ”duck” images and sounds. The mutual information generated one cluster, while other neurons distributes uniformly in the SOM spaces.

and the number of iteration was 200. Figure 7 shows the result. We find the gathering of the neurons that fire correspondingly to input data of number 1–20. The mutual information between the visual and audio data sets caused the meaningful clustering. We can interpret this self-organization as generation of concept of duck.

## 4 Summary

We proposed the concept generation method based on the mutual information quantity among multiple self-organization maps. Experiments demonstrated that the method works effectively to make clusters, which can be interpreted as concept generation. With this method, we can treat data in various modes besides visual and audio signals. The number of the modes is unlimited. This method possesses high scalability.

## References

1. Weizenbaum, J.: ELIZA - A computer program for the study of natural language communication between man and machine. Communications of the ACM 9(1), 36–45 (1966)
2. Winograd, T., Flores, F.: Understanding Computers and Cognition: A New Foundation for Design. Addison-Wesley, Reading (1987)
3. Kozima, H.: Infanoid: A babybot that explores the social environment. In: Dautenhahn, K., Bond, A.H., Canamero, L., Edmonds, B. (eds.) Socially Intelligent Agents: Creating Relationships with Computers and Robots, pp. 157–164. Kluwer Academic Publishers, Dordrecht (2002)
4. Iwahashi, N.: Robots that learn language – developmental approach to human-machine conversations. In: Proceeding of International Workshop on Emergence and Evolution of Linguistic Communication, pp. 142–179 (2006)
5. Ramachandran, V., Blakeslee, S., Sacks, O.: Phantoms in the Brain: Probing the Mysteries of the Human Mind. Harper Perennial (1999)
6. Roy, D., Pentland, A.: Learning words from sights and sounds: a computational model. Cognitive Science 26(1), 113–146 (2002)
7. Roy, D., Mukherjee, N.: Towards situated speech understanding: Visual context priming of language models. Computer Speech and Language 19(2), 227–248 (2005)
8. Kohonen, T., Kaski, S., Lappalainen, H.: Self-organized?formation of various invariant-feature in the adaptivesubspace? SOM. Neural Computation 9, 1321–1344 (1997)
9. Kohonen, T.: Generalization of the self-organizing map. In: Proceeding of International Joint Conference on Neural Networks, pp. 457–462 (1993)
10. Otsu, N., Kurita, T.: A new scheme for practical, flexible and intelligent vision systems. In: Proceeding of IAPR Workshop on Computer Vision, pp. 431–435 (1988)
11. Young, S., Everman, G., Hain, T., Kershaw, D., Moore, G., Odel, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: The HTK Book. Cambridge University Engineering Department (1995)

# Decoding Ambisonic Signals to Irregular Loudspeaker Configuration Based on Artificial Neural Networks

Peter Wai-Ming Tsang, Wai Keung Cheung, and Chi Sing Leung

Dept. of Electronic Engineering, City University of Hong Kong, Hong Kong

**Abstract.** This paper present a novel scheme which applies Artificial Neural Network (ANN) for determining the decoding parameters of a first order ambisonic system so that the three dimensional sound field can be reconstructed with an arbitrary quad speaker configuration. Differ from approaches based on the Modified Tabu Search (MTS) and the Heuristic Genetic Algorithm (HGA) which involve large number of iterations, the proposed method provides a significant reduction in computation. Experimental evaluation demonstrates that our method is significantly faster than existing schemes, and also exhibit higher accuracy and stability than the MTS.

## 1 Introduction

Ambisonic is a technology developed by Gerson [1,2,3,4] and Fellgett [5]to encoded a three dimensional sound field into a multiple channel data sequence known as the B-format. Theoretically, through a plurality of loudspeakers [5] with arbitrary spatial distribution, the encoded B-format signal can be used to reconstruct the original sound field with a decoder. One of the popular setups is the quad rig formed by four loudspeakers each positioned at a unique quadrant of a circle.

Early ambisonic systems employ simple loudspeakers configurations such as the square or the rectangular arrays with pre-computed sets of decoding parameters. Irregular arrays of $M$ ($M$ being three or more) pairs of diametrically opposite loudspeakers have also been addressed [6]. Once these parameters are available, the B-format data can be easily reverted to the sound field with numerical means [7]. However for arbitrary loudspeaker rigs, the determination of the decoding parameters becomes an NP problem. Recently, Wiggins [8,9] have adopted the Modified Tabu Search (MTS) algorithm to address this problem. Later, an alternative attempt has been made by Tsang [10] who employs Heuristic Genetic Algorithm (HGA) to optimize the decoding parameters, resulting in more stable and accurate estimation of the decoding parameters. However, the optimization process involves large number of iterations and the computation time is lengthy. Hence, once the speaker positions are changed, a time-consuming optimziation is required for search a new set of decoding parameters.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 273–280, 2009.

In this paper, we propose to use an neural network approach to build the mapping from speaker positions to the decoding parameters. To achieve that we first trained an ANN with different corresponding pairs of loudspeaker distribution and optimal set of decoding parameters, derived from the HGA scheme. Subsequently when an arbitrary loudspeaker configuration is presented, it is input to the ANN and the set of decoding parameters is retrieved from the output layer of the neural network.

The organization of this paper is as follows. In Section 2, we review the concept of ambisonic and some existing approach. In Section 3, the ANN approach to Decoding ambisonic signals to irregular loudspeaker configuration is presented. Section 4 presents our simulation results. Conclusion is presented in Section 5.

## 2 Background of the Ambisonic Technology

The ambisonic technology [1,2,3,4,5] decomposes a three dimensional sound field into spherical harmonic components commonly represented with the B-format. Initially the process was conducted by direct recording the acoustic signal with the sound field microphone formed by the juxtaposition of three microphones. With the advancement of computers the B-format can now be generated with numerical means. In practical applications where cost and space are crucial factors to be considered, the implementation is usually simplified into a first order, full horizontal system to recover a planar sound field with small number of loudspeakers.

With this approximation, the first order ambisonic signal is comprised of three components, namely the sound pressure $W$, plus a pair of sound pressure gradient terms$\{X, Y\}$, which are the cylindrical decomposition of the sound field. Given an input signal $s$ located on the unit circle and at an azimuth angle $\theta$ (encoding angle), the conversion is given by

$$\left[ W \; X(\theta) \; Y(\theta) \right] = s \left[ \mathrm{sqrt}\{2\}/2 \cos(\theta) \sin(\theta) \right] . \tag{1}$$

Early ambisonic systems assume restricted set of simple set of loudspeakers configurations so that the decoders can be implemented with pre-computed parameters. One of the popular setups is the quad rig formed by four loudspeakers each positioned at a unique quadrant of a circle. The sound field, neglecting frontal dominance, can be decoded from the B-format signals by setting the gain of the four loudspeakers as

$$\begin{bmatrix} g_1(\theta) \\ g_2(\theta) \\ g_3(\theta) \\ g_4(\theta) \end{bmatrix} = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \\ k_7 & k_8 & k_9 \\ k_{10} & k_{11} & k_{12} \end{bmatrix} \begin{bmatrix} W \\ X(\theta) \\ Y(\theta) \end{bmatrix} . \tag{2}$$

For some simple and symmetrical quad configurations the decoding parameters $k_i \mid_{1 \le i \le 12}$ have been determined and made known in many sources.

To handle arbitrary loudspeaker rigs determination of the decoding parameters become an NP problem. Recently, Wiggins [8,9] have adopted the MTS algorithm to address this problem by optimizing an Aggregate Objective Function (AOF) that is formulated based on the Velocity vector $\overrightarrow{R_V}\ (\theta)$and the Energy vector $\boldsymbol{R}_E\,(\theta)$. Both quantities are derived from the six terms, given by

$$P\,(\theta) = \sum_{i=1}^{N} g_i\,(\theta)\,, \qquad E\,(\theta) = \sum_{i=1}^{N} g_i\,(\theta)^2$$

$$E_X\,(\theta) = \sum_{i=1}^{N} g_i\,(\theta)^2 \cos(\theta_i)/E\,(\theta)\,,\; E_Y\,(\theta) = \sum_{i=1}^{N} g_i\,(\theta)^2 \sin(\theta_i)/E\,(\theta)$$

$$V_X\,(\theta) = \sum_{i=1}^{N} g_i\,(\theta) \cos(\theta_i)/P\,(\theta)\,,\;\; V_Y\,(\theta) = \sum_{i=1}^{N} g_i\,(\theta) \sin(\theta_i)/P\,(\theta)$$

where $N$ is the total number of loudspeakers in a quad rig, $\theta_i$ represents the angular position of the $i$-th loudspeaker. From the above expressions, Velocity and Energy vectors are derived as

$$\begin{aligned}
r_V\,(\theta) &= |\boldsymbol{R}_V\,(\theta)| = \sqrt{V_X\,(\theta)^2 + V_Y\,(\theta)^2} \\
\theta_V\,(\theta) &= \arg\left(\boldsymbol{R}_V\,(\theta)\right) = \arctan(V_Y\,(\theta)\,/V_X\,(\theta)) \\
r_E\,(\theta) &= |\boldsymbol{R}_E\,(\theta)| = \sqrt{E_X\,(\theta)^2 + E_Y\,(\theta)^2} \\
\theta_E\,(\theta) &= \arg\left(\boldsymbol{R}_E\,(\theta)\right) = \arctan(E_Y\,(\theta)\,/E_X\,(\theta))\,.
\end{aligned} \tag{3}$$

The AOF is a term to reflect how well the sound field is reconstructed with respect to the decoding parameters. For low frequency below 700Hz this is formed by the sum of three fitness values given by

$$F_L = V_{Fit} + 0.8M_{Fit} + A_{Fit} \tag{4}$$

where $V_{Fit}$, $M_{Fit}$, and $A_{Fit}$ are given by

$$\begin{aligned}
V_{Fit} &= \sqrt{\frac{1}{360}\sum_{\theta=0}^{360}\left[1 - \frac{P\,(0)}{P\,(\theta)}\right]^2} \\
M_{Fit} &= \sqrt{\frac{1}{360}\sum_{\theta=0}^{360}\left[1 - r_V\,(\theta)\right]^2} \\
A_{Fit} &= \sqrt{\frac{1}{360}\sum_{\theta=0}^{360}(\theta - \theta_V\,(\theta))^2} + \sqrt{\frac{1}{360}\sum_{\theta=0}^{360}(\theta - \theta_E\,(\theta))^2}\,.
\end{aligned} \tag{5}$$

The optimal solutions correspond to decoding parameters which result in $F_L$=0 implying uniform distribution of the Energy and Velocity vectors for all encoding angles (i.e., portraying a circular locus). Here we shall only restricted our analysis to low frequency signals as similar treatment can be extended to high frequency ones. Deducing the twelve decoding parameters to optimize the AOF is a hard problem and a simplification has been made by reducing the number of parameters to six by revising the formulation as

$$\begin{bmatrix} g_1(\theta) \\ g_2(\theta) \\ g_3(\theta) \\ g_4(\theta) \end{bmatrix} = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_1 & k_2 & -k_3 \\ k_4 & -k_5 & k_6 \\ k_4 & -k_5 & -k_6 \end{bmatrix} \begin{bmatrix} W \\ X(\theta) \\ Y(\theta) . \end{bmatrix} \tag{6}$$

Although successful demonstration has been made in reconstructing horizontal sound fields on a 5 loudspeaker configuration, the MTS is sensitive to the initial setting of the parameters and does not perform equally well with less number of loudspeakers. A poor estimation on the starting condition can lead to large error in the decoding parameters, hence reducing the fidelity of the reconstructed sound field. Later, an alternative attempt has been made by Tsang [10] who employs HGA to optimize the AOF, resulting in more stable and accurate estimation of the decoding parameters. Despite the favorable outcome of these two methods, the optimization process involves large number of iterations and the computation time is lengthy.

## 3 Estimation of Decoding Parameters Based on Artificial Neural Networks (ANN)

In this paper we shall restrict our analysis to a quad rig with arbitrary loudspeaker position in each quadrant. The ambisonic decoder can be modelled as a system with the angular positions of the loudspeakers, denoted by $A = [\theta_1, \theta_2, \theta_3, \theta_4]$ and the decoding parameters $K = [k_1, k_2, k_3, k_4, k_5, k_6]$ as the input and output vectors, respectively, i.e. $K = F(A)$. The mapping between correspondence pairs of input and output vectors are unknown and we only have some sampling input-output pairs (with the optimized AOF) obtained by other methods.

We propose that the function $F(A)$ can be realized with an ANN which is trained by associating an example set of loudspeaker configuration with their

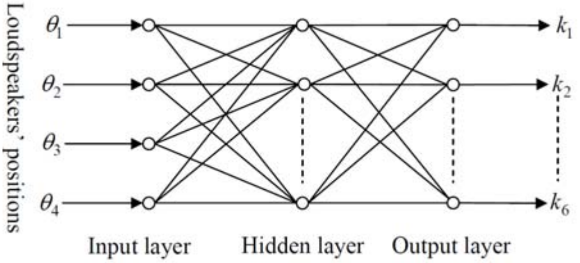


**Fig. 1.** ANN for associating the decoding parameters with the loudspeakers' positions

optimal sets of decoding parameters. To study the feasibility of the approach the classical three layered Backpropagation Network (BPN) [11] is adopted and depicted in Figure 1.

The input layer consists of four neurons each accepting an element of the position vector. The output layer consists of six neurons each corresponding to one of the decoding parameters. The network is trained with the Levenberg-Marquardt [12]. After training, the set of decoding parameters $K$ for an arbitrary configuration $A$ is obtained by applying the latter to the ANN.

## 4 Experimental Results

To evaluate the performance of our method different sets of loudspeaker configuration listed in Table 1 are input to the ANN with and in each case the corresponding set of decoding parameters is obtained from the activation of the output neurons. The AOF value, velocity and energy vectors are computed and the latter two are plotted for angular positions ranging from 0° to 360°. These results are compared against that derived from the use of HGA and MTS in determining the decoding parameters.

**Table 1.** Loudspeaker configurations for training the ANN

| | | | |
|---|---|---|---|
| 35° 125° 215° 305° | 55° 125° 215° 305° | 30° 120° 210° 300° | 60° 120° 210° 300° |
| 35° 125° 215° 325° | 55° 125° 215° 325° | 30° 120° 210° 330° | 60° 120° 210° 330° |
| 35° 125° 235° 305° | 55° 125° 235° 305° | 30° 120° 240° 300° | 60° 120° 240° 300° |
| 35° 125° 235° 325° | 55° 125° 235° 325° | 30° 120° 240° 330 | 60° 120° 240° 330° |
| 35° 145° 215° 305 | 55° 145° 215° 305° | 30° 150° 210° 300° | 60° 150° 210° 300° |
| 35° 145° 215° 325° | 55° 145° 215° 325° | 30° 150° 210° 330° | 60° 150° 210° 330° |
| 35° 145° 235° 305° | 55° 145° 235° 305° | 30° 150° 240° 300° | 60° 150° 240° 300° |
| 35° 145° 235° 325° | 55° 145° 235° 325° | 30° 150° 240° 330° | 60° 150° 240° 330° |

Typical results can be shown with an irregular out-training-set quad loudspeaker configuration [$40^o$ $140^o$ $215^o$ $305^o$]. The optimal solution obtained with HGA after 100 generations (based on a population size of 100) is given in Figure 2a. The AOF value is 0.174 and the distributions of the Energy and Velocity vectors are rather uniform. The result obtained with our method is shown in Figure 2b. It can be seen that the performance is close to HGA. Although the AOF value is slightly lower the amount of computation is significantly smaller, reflecting the feasibility of our approach and its potential in practical applications.

To further evaluate the feasibility of the proposed neural network enumerator (NNE) approach, we compare it against the HGA method and NNE with MTS. In the NNE-MTS, we the initial guess of MTS is selected from the result of NNE. We employ identical test set of quad loudspeaker configuration that has been adopted in [10] as listed in Table 2.

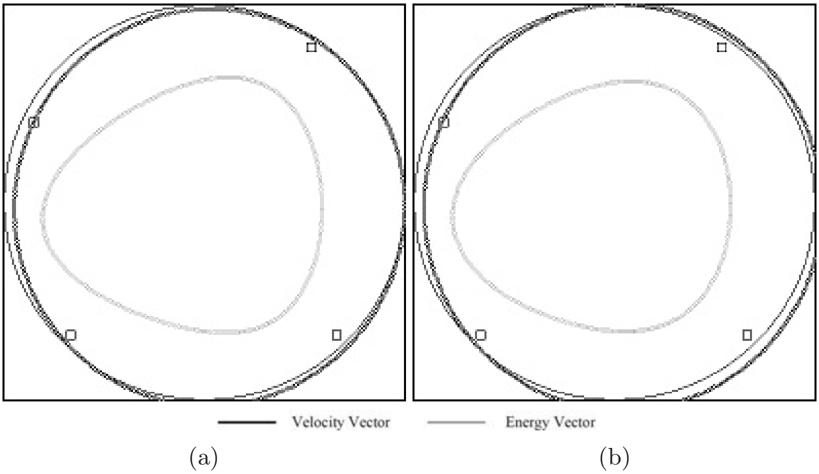


**Fig. 2.** Typical result of the reconstructed Velocity and Energy Vectors. (a) Results obtained with HGA for low frequency signals after 100 generations ( $F_L = 0.174$). (b) Results obtained with ANN for low frequency signals ( $F_L = 0.177$).

**Table 2.** Loudspeaker configurations for testing the performance

| Config. | Loudspeaker positions | Config. | Loudspeaker positions |
|---|---|---|---|
| 1 | $45^o$, $135^o$, $225^o$, $315^o$ | 6 | $30^o$, $135^o$, $235^o$, $315^o$ |
| 2 | $30^o$, $110^o$, $250^o$, $330^o$ | 7 | $45^o$, $135^o$, $205^o$, $305^o$ |
| 3 | $60^o$, $120^o$, $240^o$, $300^o$ | 8 | $60^o$, $135^o$, $205^o$, $310^o$ |
| 4 | $30^o$, $135^o$, $225^o$, $315^o$ | 9 | $60^o$, $135^o$, $245^o$, $350^o$ |
| 5 | $45^o$, $125^o$, $225^o$, $315^o$ | 9 | $50^o$, $150^o$, $220^o$, $290^o$ |

To begin with the HGA is applied to determine the values of the six decoding parameters for low frequency ranges in each test configuration. In order to ensure accurate deduction of decoding parameters 100 generations of evolution are conducted on a population consisting of 100 chromosomes. The complexity is equivalent to 10,000 iterations in the MTS scheme. Next the estimated decoding parameters for each case are obtained with NNE and NNE-MTS (with 100 iterations of MTS) with the neural network trained with the samples given in Table I. The AOF values obtained with the three methods are shown in Figure 3. It can be seen that NNE-MTS results in AOF values which are close to HGA, and generally lower than or equal to NNE. From the figure, the NNE can be produce a very satisfory results. Besides, the results can be further optimized by the MTS algorithm with a very small number of iteration.

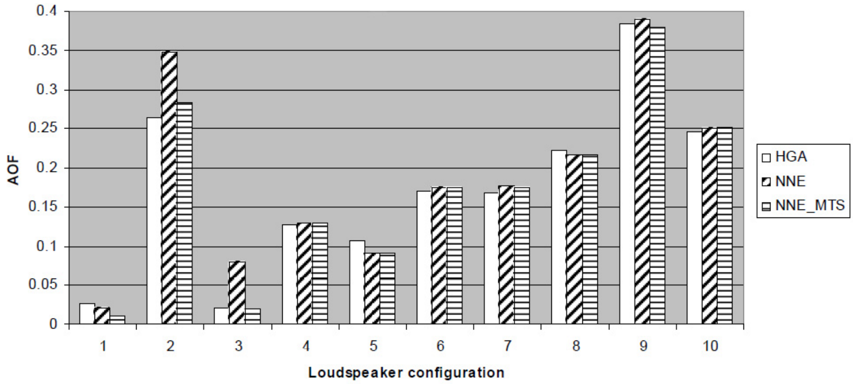


**Fig. 3.** Results of $F_L$ of various approaches

## 5 Conclusion

Past researches have demonstrated the feasibility of reconstructing, with irregular loudspeaker configuration, horizontal planar sound field from first order B-format ambisonic signal. The result enables surround sound signals to be played back under circumstances where loudspeakers positions have to be changed according to the spatial constraint of the housing environment. However, with existing methods, such as HGA or MTS, the computational complexity is quite large. This paper proposed an ANN approach to determine the decoding parameters to play back the surround sound signals. Unlike the conventional HGA or MTS, where a time consuming algorithm is required to tune the decoding parameters when a new setting is given, our scheme are significantly faster than the HGA or the MTS methods. We anticipated that the works in this paper could also benefit further research in related areas. For example, it can be extended to reconstruct higher order ambisonic signals or it can used to construct Holograpic sound systems.

## References

1. Gerzon, M.A.: Surround-sound from 2-channel stereo. HiFi News (August 1970)
2. Gerzon, M.A.: Surround-sound psychoacoustics. Wireless World (December 1974)
3. Gerzon, M.A.: Multidirectional sound reproduction systems. United States Patent, 3,997,725 (December 1976)
4. Gerzon, M.A.: Ambisonics in multichannel broadcasting and video. J. Audio Eng. Soc. 33, 859–871 (1985)
5. Fellgett, P.: Ambisonics. Part One: General system description. Studio Sound, 20–40 (August 1975)
6. Gerzon, M.A.: Decoders for feeding irregular loudspeaker array. United States Patent 4,414,430 (November 1983) [10]

7. Farina, A.: Software Implementation of B-Format Encoding and Decodin. In: Proc. of the 104-th AES Conv. Amsterdam (1998)
8. Wiggins, B., et al.: The Design and Optimization of Surround Sound Decoders using Heuristic Methods. In: Proc. UKSim 2003, pp. 106–114 (2003)
9. Wiggins, B.: An Investigation into the Real-time Manipulation and Control of Three-dimensional Sound Fields, PhD thesis, University of Derby, Derby, UK (2004)
10. Tsang, P.W.M., Cheung, K.W.K.: Development of a Re-configurable Ambisonic Decoder for Irregular Loudspeaker Configuration. IET Cir. Dev. Syst. (in press, 2009)
11. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, pp. 318–368 (1986)
12. Hagan, M.T., Menhaj, M.B.: Training feedforward networks with the Marquardt algorithm. IEEE Trans. Neural Networks 5(6), 989–993 (1994)

# Document Clustering with Cluster Refinement and Non-negative Matrix Factorization

Sun Park[1], Dong Un An[2,*], ByungRea Char[3], and Chul-Won Kim[4]

[1] Advanced Graduate Education Center of Jeonbuk for Electronics and Information Technology-BK21, Chonbuk National University, Korea
sunbak@jbnu.ac.kr
[2] Division of Electronic & Infomation Engineering, Chonbuk National University, Korea
duan@chonbuk.ac.kr
[3] Network Media Lab., GIST, Korea
brcha@nm.gist.ac.kr
[4] Department of Computer Engineering, Honam University, Korea
cwkim@honam.ac.kr

**Abstract.** Document clustering is an important method for document analysis and is used in many different information retrieval applications. This paper proposes a new document clustering method using the clustering method based NMF (Non-negative Matrix Factorization) and refinement of documents in clusters by using coherence of cluster. The proposed method can improve the quality of document clustering because the re-assigned documents in cluster by using coherence of cluster based similarity between documents, the semantic feature matrix and the semantic variable matrix, which is used in document clustering, can represent an inherent structure of document set better. The experimental results demonstrate that the proposed method achieves better performance than other document clustering methods.

## 1 Introduction

The explosive increase in internet access has produced a large amount of information, and caused the problem that many documents with the same or similar topics were duplicated. This kind of data duplication problem increases the necessity for effective document clustering methods. Also, document clustering has been receiving more and more attentions as an important method for unsupervised document organization, automatic summarization, topic extraction, and information filtering or retrieval [1, 2, 3, 11, 13].

A clustering technique is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters [1, 2, 3, 11].

Generally, document clustering method can be classified into hierarchical methods and partitioning methods [1, 3]. Hierarchical clustering methods proceed successively by building a tree of clusters. On the other hand, partitioning clustering methods decompose the data set into *k* disjoint classes such that the data points in a class are nearer to one another than the data points in other classes [1, 3, 10].

[*] Corresponding author.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 281–288, 2009.

Recent studies for document clustering method use graphs [4, 13], machine learning [5, 9 17], and matrix factorization [15, 16]. Graphs based methods model the given document set using an undirected graph in which each node represents a document [4, 13]. Machine learning based methods use semi-supervised clustering model with respect to prior knowledge and documents' membership [5, 9 17]. Matrix factorization based method use semantic features of documents set for document clustering [15, 16].

The NMF (Non-negative Matrix Factorization) can represent individual object as the non-negative linear combination of part information extracted from a large volume of objects. This method can deal with a large volume of information efficiently since original non-negative matrix is decomposed into sparsely distributed representation of two non-negative matrices [6, 7, 14]. Advantages of NMF, The NMF have a great power to easily extract semantic features representing the inherent structure of data objects. The factorization result of NMF has better semantic interpretation and the clustering result can be easily derived from it [15, 16]. However, there is still some disadvantage of NMF, it might limit successful decomposing semantic features from any data set as data objects viewed from extremely different viewpoints, or highly articulated objects [6].

In this paper, we propose a document clustering method using NMF clustering and refinement clustering by coherence of cluster. Cluster coherence is used to measure the density between documents in cluster. It means how close the similarity between documents and major topics of cluster.

The proposed method has the following advantages. First, it can group important documents in document set using semantic variables by NMF. So it can identify major topics and subtopics of clusters with respect to the semantic variable by NMF. Second, it can remove the dissimilar documents in clusters using cluster coherence. So, it can improve the quality of document clustering since refinement cluster helps us to remove dissimilarity information easily and avoid biased inherent semantics of documents to be reflected in clusters by NMF.

The rest of the paper is organized as follows: Section 2 describes the related works regarding document clustering methods. In Section 3, we describe the NMF algorithm in detail. In Section 4, the proposed document clustering method is introduced. Section 5 shows the evaluation and experimental results. Finally, we conclude in Section 6.

## 2 Related Works

Traditional clustering method can be classified into partitioning, hierarchical, density-based, and grid-based. Most of these methods use distance functions as object criteria and are not effective in high dimensional spaces [1, 3, 10].

Wang and Zhang proposed the document clustering with local and global regularization (CLRG). It uses a local label predictors and a global label smoothness regularizer [13]. Liu et al. proposed a document clustering method using cluster refinement and model selection. It uses GMM (Gaussian Mixture Model) and EM (Expectation Maximization) algorithm to conduct an initial document clustering. It also refines the initially obtained document clusters by voting on the cluster label of each document [9]. Ji et al. proposed a semi-supervised clustering model that incorporates prior knowledge about documents' membership for document clustering analysis [5]. Zhang et al. adopt a RL (Relaxation Labeling) based cluster algorithm to evaluate the effectiveness of the aforementioned types of links for document clustering. It use both content and linkage information in dataset [17]. Xu et al proposed a document partitioning method based on the non-negative factorization matrix (NMF) of the given document corpus [16]. Xu and Gong proposed a data clustering method models each cluster as a linear combination of the data points, and each data point as a linear combination of the cluster centers [15]. Li and Ding presented an overview and summary on various matrix factorization algorithms for clustering and theoretically analyze the relationships among them [8].

Li et al. proposed a document clustering algorithm ASI (adaptive subspace iteration) using explicitly modeling of the subspace structure associated with each cluster [10]. Wang et al. proposed a clustering approach for clustering multi-type interrelated data objects. It fully explores the relationship between data objects for clustering analysis [12].

## 3 Non-negative Matrix Factorization

In this paper, we define the matrix notation as follows: Let $X_{*j}$ be $j$'th column vector of matrix $X$, $X_{i*}$ be $i$'th row vector and $X_{ij}$ be the element of $i$'th row and $j$'th column.

Non-negative matrix factorization (NMF) is to decompose a given $m$×$n$ matrix $A$ into a non-negative semantic feature matrix $W$ and a non-negative semantic variable matrix $H$ as shown in Equation (1).

$$A \approx WH \tag{1}$$

where $W$ is an $m \times r$ non-negative matrix and $H$ is an $r \times n$ non-negative matrix. Usually $r$ is chosen to be smaller than $m$ or $n$, so that the total sizes of $W$ and $H$ are smaller than that of the original matrix $A$.

We use the objective function that minimizes the Euclidean distance between each column of $A$ and it approximation $\tilde{A} = WH$ , which was proposed in Lee and Seung [9, 10]. As an objective function, the Frobenius norm is used [14]:

$$\Theta_E(W,H) \equiv \|A - WH\|_F^2 \equiv \sum_{j=1}^{m}\sum_{i=1}^{n}\left( X_{ji} - \sum_{l=1}^{r} W_{jl}H_{li} \right)^2 \tag{2}$$

We keep updating $W$ and $H$ until $\Theta_E(W,H)$ converges under the predefined threshold or exceeds the number of repetition. The update rules are as follows:

$$H_{\alpha\mu} \leftarrow H_{\alpha\mu} \frac{(W^T A)_{\alpha\mu}}{(W^T WH)_{\alpha\mu}}, W_{i\alpha} \leftarrow W_{i\alpha} \frac{(AH^T)_{i\alpha}}{(WHH^T)_{i\alpha}} \tag{3}$$

The powers of the two non-negative matrices $W$ and $H$ are described as follows: All semantic variables ($H_{lj}$) are used to represent each sentence. $W$ and $H$ are represented sparsely. Intuitively, it make more sense for each sentence to be associated with some small subset of a large array of topics ($W_{*l}$), rather than just one topic or all the topics. In each semantic feature ($W_{*l}$), the NMF has grouped together semantically related terms. In addition to grouping semantically related terms together into semantic features, the NMF uses context to differentiate between multiple meanings of the same term [6].

# 4 The Proposed Document Clustering Method

In this section, we propose a method that clusters documents by cluster refinement based coherence of cluster and NMF clustering. The proposed method consists of the preprocessing phase, document clustering phase, and the cluster refinement phase. We will give a full explanation of three phases as Figure 1.

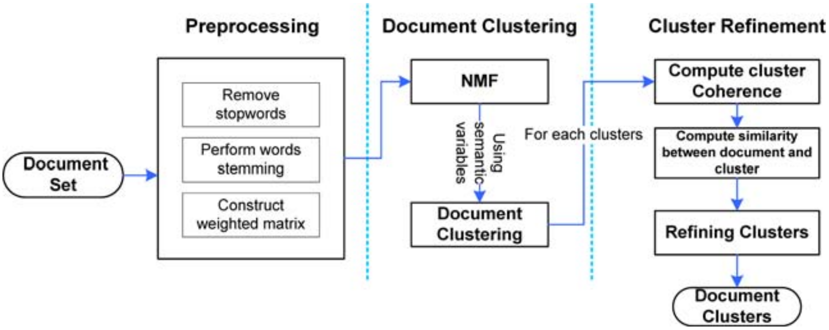


**Fig. 1.** Document clustering method using cluster refinement and NMF clustering

## 4.1 Preprocessing

In the preprocessing phase, we remove all stop-words by using Rijsbergen's stop-words list and perform words stemming by Porter's stemming algorithm [2, 11]. Then we construct the weighted term-frequency vector for each document in the document set by Equation (4) [1, 2, 11, 14].

Let $T_i = [t_{1i}, t_{2i} \dots t_{ni}]^T$ be the term-frequency vector of document $i$, where elements $t_{ji}$ denotes the frequency in which term $j$ occurs in document $i$. Let $A$ be $m \times n$ weighted terms by documents matrix, where $m$ is the number of terms and $n$ is the number of documents in a document set. Let element $A_{ji}$ be the weighted term-frequency of term $j$ in a document $i$.

$$A_{ji} = L(j, i) \cdot G(j, i) \tag{4}$$

where $L(j, i)$ is the local weighting for term $j$ in document $i$, and $G(j, i) = \log(N/n(j))$ is the global weighting for term $j$ in the whole documents.

## 4.2 Document Clustering by NMF

In the document clustering phase, document clustering uses NMF clustering method to cluster a document into different cluster label by the Xu's document clustering method using NMF [16]. We perform preprocessing phase, let $r$ be the number of cluster label. We then perform the NMF on $A$ to obtain the two non-negative matrixes $W$ and $H$ by equation (3). We use matrix $H$ to determine the cluster label of each data point. We assign sentence $A_{*i}$ to cluster $x$ if $x = \arg\max_{j} H_{ji}$.

We assume that $f$ clusters are constructed and then weighted matrix $A$ can be represented as follows:

$$\{A_{*i} \mid i=1,...,n\} = \bigcup_{k=1}^{f}\{C_l^k \mid l=1,...,s_k\}, C^p \cap C^q \neq \phi, p \neq q \tag{5}$$

where $C^k$ is the weighted matrix of $k$'th the cluster of documents, $s_k$ is the number of column of $C^k$, or the number of document in cluster $C^k$, $n$ is the number of documents, and $f$ is the number of clusters.

### 4.3 Refining Clusters by Coherence of Cluster

The cluster refinement phase is described as follows. First, The coherence of cluster $c(C^k)$ is calculated. The cluster coherence use similarity between documents in cluster $C^k$ as shown in Equation (6).

$$c(C^k) = \frac{\sum_{A_{*a}\in C^k}\left(\sum_{A_{*b}\in C^k-\{A_{*a}\}} c\sin(A_{*a}, A_{*b})\right)}{s_k} \tag{6}$$

where $k$ is the number of cluster label, *csin*() is a cosine similarity function. The coherence of cluster means how many the documents in cluster reflects major topics of cluster. The cosine similarity function *csin*() between the vector of $a$'th sentence $A_{*a}$ and the vector of $b$'th sentence $A_{*b}$ is computed as follows [2, 11].

$$c\sin(A_{*a}, A_{*b}) = \frac{A_{*a}\cdot A_b}{|A_{*a}|\times|A_{*b}|} = \frac{\sum_{j=1}^{m} A_{ja}\times A_{jb}}{\sqrt{\sum_{j=1}^{m} A_{ja}^2}\times\sqrt{\sum_{j=1}^{m} A_{jb}^2}} \tag{7}$$

Second, the cosine similarity between documents $C_l^k$ and cluster $C^k$ is calculated. Finally, the documents are re-assigned to cluster by cluster coherence and document similarity.

### 4.4 The Proposed Document Clustering Algorithm

The proposed document clustering algorithm using both the refinement clusters and NMF clustering is as follows:

**Algorithm.** Document Clustering by refinement cluster and NMF

**Input:** The term frequency matrix $A$, the number of cluster $r$, the number of whole document $n$

**Output:** re-assigned cluster $C^k$

```
1. Perform preprocessing phase
2. Obtain the non-negative matrix $W$ and $H$ from the matrix $A$ by equation (3).
3. Repeat
4.     Assign document $A_{*i}$ to cluster $C^k$ if $C^k = \arg\max_{1\le j\le r} H_{ji}$
5. Until $i = n$
6. Calculate each coherence of cluster $c(C^k)$ by equation (7), $k$=1, …, $r$
```

```
7. Perform the following steps for each cluster $C^k$.
8.     Repeat
9.         Compute $c\sin(A^k_{*i}, C^k)$ by equation (7).
10.        Put the document $A^k_{*i}$ into a candidate set $S$ if $c\sin(A^k_{*i}, C^k) < c(C^k)$
11.    Until $k = r$.
12.    Repeat
13.        Assign document $S_{*i}$ to cluster $C^k$ if $C^k = \arg\max_{1\le k\le r}(c\sin(S_{*i}, C^k))$
14.    Until $i = p$
```

where $p$ is the number of candidate document. In step 2 to 5, document clustering phase uses Xu's document clustering method using NMF. In step 6 to 14, cluster refinement phase uses coherence of cluster and document similarity.

## 5 Performance Evaluations

We use the *Reuters*[1] document corpora to evaluate the proposed method. *Reuters* corpus have 21578 documents which are grouped into 135 clusters. Many documents in Reuters have multiple category labels and documents in each cluster have a broader variety of content [13]. As an experimental data, we use mixing documents from multiple clusters randomly selected from the *Reuters* document corpus. At each run of the experiment, documents from selected number $k$ of cluster label are applied to the clustering process. The result is evaluated by comparing the cluster label of each document with its label provided by the *Reuters* document corpus.

We use the normalized mutual information metric $\overline{MI}$ for measuring the document clustering performance [13, 15, 16]. Given the two sets of document clusters $C$, $C'$, their mutual information metric $MI(C,C')$ is defined as equation (8)

$$MI(C,C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot \log_2 \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)} \tag{8}$$

Where $p(c_i)$, $p(c'_j)$ denote the probabilities that a document arbitrarily selected from the corpus belongs to the clusters $c_i$, $c'_j$, respectively, and $p(c_i, c'_j)$ denotes the joint probability that this arbitrarily selected document belongs to the cluster $c_i$ as well as $c'_j$ at the same time. $MI(C,C')$ takes values between zero and $\max(H(C),H(C'))$, where $H(C)$ and $H(C')$ are the entropies of $C$ and $C'$, respectively. It does not need to find the corresponding counterpart in $C'$, and the value keeps the same for all kinds of permutations. We use the normalized metric $\overline{MI}$ which takes values between zero and one as equation (9) [13, 15, 16].

$$\overline{MI}(C,C') = \frac{MI(C,C')}{\max(H(C), H(C'))} \tag{9}$$

[1] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

We have conducted performance evaluation by testing proposed method and comparing it with 4 other representative data clustering method using the same data corpora. We implemented 5 document clustering methods: RNMF, KM, NMF, ASI, and CLRG. In figure 2, RNMF denotes our proposed method. KM denotes partitioning method using traditional *k*-means [1, 3]. NMF demotes Xu's method using non-negative matrix factorization [16]. ASI denotes Li's method using adaptive subspace iteration [10]. CLRG denotes Wang's method using local and global regularization [13].

The evaluation results are shown in figure 2. The evaluations were conducted for the cluster numbers ranging from 2 to 10. For each given cluster number *k*, 50 experiment runs were conducted on different randomly chosen clusters, and the final performance values were obtained by averaging the values from the 50 experiments.

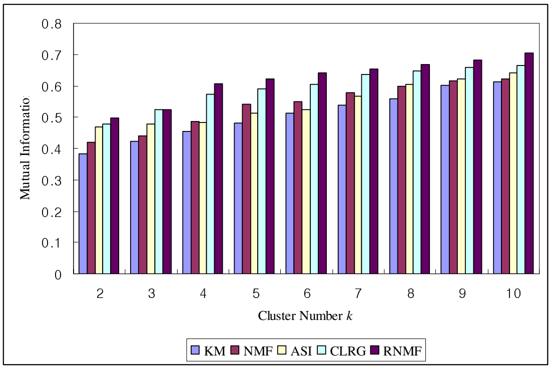


**Fig. 2.** Evaluation results of performance comparison using the average $\overline{MI}$

In figure 2, the average normalized metric $\overline{MI}$ evaluation results of RNMF is approximately 18.56% higher than that of KM, 13.39% higher than that of NMF, 12.57% higher than that of ASI, and 4.11% higher than that of CLRG. In this experiment, The RNNM showed the best performance because it uses clustering based the semantic variable and refining clusters that can identify a major topic and subtopics of documents more successfully than other methods.

## 6 Conclusions

In this paper, we have presented a document clustering method using the semantic variable based on NMF clustering and refinement of documents in cluster by coherence of cluster. The proposed method in this paper has the following advantages. First, it can remove the noise in given documents by refinement clusters. So, it can improve the quality of document clustering since coherence of cluster helps us to remove redundant information easily and avoid biased inherent semantics of documents to be reflected in clusters by NMF. Second, it can easily group documents are well covered with the major topics of document using the clustering based semantic variable by NMF. Experimental results show that the proposed method outperforms four different summarization methods.

## References

1. Chakrabarti, S.: Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufmann, San Francisco (2003)
2. Frankes, W.B., Ricardo, B.Y.: Information Retrieval, Data Structure & Algorithms. Prentice-Hall, Englewood Cliffs (1992)
3. Han, J., Kamber, M.: Data Mining Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2006)
4. Hu, T., Xiong, H., Zhou, W., Sung, S.Y., Luo, H.: Hypergraph Partitioning for Document Clustering: A Unified Clique Perspective. In: Proceeding of SIGIR 2008, pp. 871–872 (2008)
5. Ji, X., Xu, W., Zhu, S.: Document Clustering with Prior Knowledge. In: Proceeding of SIGIR 2006, pp. 405–412 (2006)
6. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
7. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems 13, 556–562 (2001)
8. Li, T., Ding, C.: The Relationships Among Various Nonegative Matrix Factorization Method for Clustering. In: Proceeding of ICDM 2006 (2006)
9. Liu, X., Gong, Y., Xu, W., Zhu, S.: Document Clustering with Cluster Refinement and Model Selection Capabilities. In: Proceeding of SIGIR 2002, pp. 191–198 (2002)
10. Li, T., Ma, S., Ogihara, M.: Document Clustering via Adaptive Subspace Iteration. In: Proceeding of SIGIR 2004, pp. 218–225 (2004)
11. Ricardo, B.Y., Berthier, R.N.: Moden Information Retrieval. ACM Press, New York (1999)
12. Wang, J., Zeng, H., Chen, Z., Lu, H., Tao, L., Ma, W.Y.: ReCoM: Reinforcement Clustering of Multi-Type Interrelated Data Objects. In: Proceeding of SIGIR 2003 (2003)
13. Wang, F., Zhang, C.: Regularized Clustering for Documents. In: Proceeding of ACM SIGIR 2007, pp. 95–102 (2007)
14. Wild, S., Curry, J., Dougherty, A.: Motivating Non-Negative Matrix Factorizations. In: Proceeding of SIAM ALA 2003 (2003)
15. Xu, W., Gong, Y.: Document Clustering by Concept Factorization. In: Proceeding of SIGIR 2004, pp. 202–209 (2004)
16. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: Proceeding of ACM SIGIR 2003 (2003)
17. Zhang, X., Hu, X., Zhou, X.: A Comparative Evaluation of Different Link Types on Enhancing Document Clustering. In: Proceeding of SIGIR 2008, pp. 555–562 (2008)

# Hierarchical Multi-view Fisher Discriminant Analysis

Qiaona Chen and Shiliang Sun

Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai, 200241, China
qnchen07@gmail.com, slsun@cs.ecnu.edu.cn

**Abstract.** Fisher discriminant analysis (FDA) has exhibited its great power to improve the performance in classification and dimensionality reduction tasks. Objects in the real world often have more than one natural feature set and therefore they often can be described by more than one views. However, traditional FDA addresses all problems with a single view. In this paper we propose multi-view FDA (MFDA) which combines traditional FDA with multi-view learning. In order to improve the performance of MFDA for multi-class case, we further propose hierarchical MFDA which combines MFDA with hierarchical metric learning. Experiments are performed on many artificial and real-world data sets. Comparisons with the single-view FDA show the effectiveness of the proposed method.

**Keywords:** Fisher discriminant analysis, multi-view learning, hierarchical metric learning.

## 1 Introduction

During the past few decades, Fisher discriminant analysis (FDA) [1] has exhibited its great power to improve performance in classification [2,3,4] and also played an important role in dimensionality reduction [5,6,7].

In real-world domains, many objects have more than one natural feature set. Take the webpage data set as an example. The class that webpage belongs to can be judged by the key words in this page and can also be judged by the key words in the hyperlinks pointing to this page. Thus many multi-view learning methods [8,9] have been proposed. Nevertheless, traditional FDA addresses problems with a single view. In this paper, we propose multi-view Fisher discriminant analysis (MFDA) which combines traditional FDA with multi-view learning. A method on multi-view Fisher discriminant analysis has been proposed in [10]. Different from ours, this method maximizes the correlation between views which is similar to canonical correlation analysis [11]. Most views' outputs may not be consistent with real labels while the correlation between views are maximal. Our proposed MFDA not only makes views be consistent with others but also makes each view's output be consistent with the real label.

There are two forms of between-class scatter in traditional FDA. One form can only be used for the two-class case. The other one can be applied to the multi-class case, but sometimes it does not work well. It is discussed in Section 3. Inspired by the recent developments on local metric learning [12,13,14,15,16,17,18], we further propose

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 289–298, 2009.

hierarchical MFDA (HMFDA), which combines hierarchical metric learning [19] with MFDA. In HMFDA, the between-class scatter is first limited to the two-class case, then hierarchical metric learning is used to extend MFDA to the multi-class case.

The remaining sections are organized as follows. Section 2 introduces our proposed MFDA in detail. Then, Section 3 exhibits HMFDA as an extension of MFDA. Section 4 demonstrates the performance of the proposed method through experiments on many data sets. Finally, Section 5 concludes the paper and discusses the direction of our future work.

## 2 Multi-view Fisher Discriminant Analysis

### 2.1 Two-View Fisher Discriminant Analysis

Considering a set of inputs $\mathbf{x}_1$,$\mathbf{x}_2$,...,$\mathbf{x}_n$ ($\mathbf{x}_i = \{x_{i1}, x_{i2}, ..., x_{id}\}$) whose labels are $y_i \in \{1, 2, ..., c\}(i = 1, 2, ..., n)$. Suppose $\mathbf{S}_{Wa}$, $\mathbf{S}_{Wb}$ are within-class-scatter matrices and $\mathbf{S}_{Ba}$, $\mathbf{S}_{Bb}$ are between-class-scatter matrices. The problem is solved by seeking two projections $\mathbf{w}_a$, $\mathbf{w}_b$ and two bias $b_a$, $b_b$:

$$\begin{aligned} \min\ & \sum_i [(y_i - \overset{\wedge}{y_{ai}})^2 + \alpha(y_i - \overset{\wedge}{y_{bi}})^2] + \sum_i [\beta(\overset{\wedge}{y_{ai}} - \overset{\wedge}{y_{bi}})^2] \\ \text{s.t.}\ & k\mathbf{w}_a\mathbf{S}_{Wa}\mathbf{w}_a^T - \mathbf{w}_a\mathbf{S}_{Ba}\mathbf{w}_a^T \leq 0 \\ & k\mathbf{w}_b\mathbf{S}_{Wb}\mathbf{w}_b^T - \mathbf{w}_b\mathbf{S}_{Bb}\mathbf{w}_b^T \leq 0. \end{aligned} \tag{1}$$

Here $\overset{\wedge}{y_{ai}}$ and $\overset{\wedge}{y_{bi}}$ are the outputs of the discriminant functions. MFDA consists of three parts. The first part is the first term of the objective function which attempts to make every input's output of discriminant function of each view be consistent with its real label. The second part is the second term of the objective function which aims to make views be consistent with others. Constant $\alpha$ and $\beta$ control the relative importance of these competing terms in the objective function. Constraint conditions (the third part), which are used to control the low range of Fisher criterion value, can be rewritten as: $\frac{\mathbf{w}_a\mathbf{S}_{Ba}\mathbf{w}_a^T}{\mathbf{w}_a\mathbf{S}_{Wa}\mathbf{w}_a^T} > k$ and $\frac{\mathbf{w}_b\mathbf{S}_{Bb}\mathbf{w}_b^T}{\mathbf{w}_b\mathbf{S}_{Wb}\mathbf{w}_b^T} > k$.

### 2.2 Quadratically Constrained Quadratic Programming Formulation

Here we bring the problem to a standard quadratically constrained quadratic programming (QCQP) form. The objection function can be reformulated as:

$$\begin{aligned} & \min\ \sum_i [(y_i - \overset{\wedge}{y_{ai}})^2 + \alpha(y_i - \overset{\wedge}{y_{bi}})^2] + \sum_i [\beta(\overset{\wedge}{y_{ai}} - \overset{\wedge}{y_{bi}})^2] \\ = & \min\ \sum_i [(y_i - \overset{\wedge}{y_{ai}})^2 + \alpha(y_i - \overset{\wedge}{y_{bi}})^2 + \beta(\overset{\wedge}{y_{ai}} - \overset{\wedge}{y_{bi}})^2] \\ = & \min\ \left(\widetilde{\mathbf{w}_a}^T\ \widetilde{\mathbf{w}_b}^T\right) \begin{pmatrix} (\beta+1)\sum_i \widetilde{\mathbf{x}_{ai}}\widetilde{\mathbf{x}_{ai}}^T & -\beta\sum_i \widetilde{\mathbf{x}_{ai}}\widetilde{\mathbf{x}_{bi}}^T \\ -\beta\sum_i \widetilde{\mathbf{x}_{bi}}\widetilde{\mathbf{x}_{ai}}^T & (\alpha+\beta)\sum_i \widetilde{\mathbf{x}_{bi}}\widetilde{\mathbf{x}_{bi}}^T \end{pmatrix} \begin{pmatrix} \widetilde{\mathbf{w}_a} \\ \widetilde{\mathbf{w}_b} \end{pmatrix} \end{aligned}$$

$$-2(\sum_i y_i \widetilde{\mathbf{x}_{ai}}^T \;\; \alpha \sum_i y_i \widetilde{\mathbf{x}_{bi}}^T)\begin{pmatrix}\widetilde{\mathbf{w}_a}\\ \widetilde{\mathbf{w}_b}\end{pmatrix}, \tag{2}$$

where

$$\overset{\wedge}{y_{ai}} = \mathbf{w}_a^T \mathbf{x}_{ai} + b_a = (\mathbf{w}_a^T \quad b_a)\begin{pmatrix}\mathbf{x}_{ai}\\ 1\end{pmatrix} \triangleq \widetilde{\mathbf{w}_a}^T \widetilde{\mathbf{x}_{ai}},$$

$$\overset{\wedge}{y_{bi}} = \mathbf{w}_b^T \mathbf{x}_{bi} + b_b = (\mathbf{w}_b^T \quad b_b)\begin{pmatrix}\mathbf{x}_{bi}\\ 1\end{pmatrix} \triangleq \widetilde{\mathbf{w}_b}^T \widetilde{\mathbf{x}_{bi}}.$$

Suppose

$$\widetilde{\mathbf{S}_{Wa}} \triangleq \sum_{j=1}^{c}\sum_{i:y_i=j}(\begin{pmatrix}\mathbf{x}_{ai}\\ 0\end{pmatrix} - \begin{pmatrix}\mathbf{m}_{aj}\\ 0\end{pmatrix})(\begin{pmatrix}\mathbf{x}_{ai}\\ 0\end{pmatrix} - \begin{pmatrix}\mathbf{m}_{aj}\\ 0\end{pmatrix})^T,$$

$$\widetilde{\mathbf{S}_{Wb}} \triangleq \sum_{j=1}^{c}\sum_{i:y_i=j}(\begin{pmatrix}\mathbf{x}_{bi}\\ 0\end{pmatrix} - \begin{pmatrix}\mathbf{m}_{bj}\\ 0\end{pmatrix})(\begin{pmatrix}\mathbf{x}_{bi}\\ 0\end{pmatrix} - \begin{pmatrix}\mathbf{m}_{bj}\\ 0\end{pmatrix})^T,$$

$$\widetilde{\mathbf{S}_{Ba}} \triangleq (\begin{pmatrix}\mathbf{m}_{a1}\\ 0\end{pmatrix} - \begin{pmatrix}\mathbf{m}_{a2}\\ 0\end{pmatrix})(\begin{pmatrix}\mathbf{m}_{a1}\\ 0\end{pmatrix} - \begin{pmatrix}\mathbf{m}_{a2}\\ 0\end{pmatrix})^T,$$

$$\widetilde{\mathbf{S}_{Bb}}^T \triangleq (\begin{pmatrix}\mathbf{m}_{b1}\\ 0\end{pmatrix} - \begin{pmatrix}\mathbf{m}_{b2}\\ 0\end{pmatrix})(\begin{pmatrix}\mathbf{m}_{b1}\\ 0\end{pmatrix} - \begin{pmatrix}\mathbf{m}_{b2}\\ 0\end{pmatrix})^T.$$

Scatters in constraints can be rewritten as:

$$\mathbf{w}_a^T \mathbf{S}_{Wa} \mathbf{w}_a \triangleq \widetilde{\mathbf{w}_a}^T \widetilde{\mathbf{S}_{Wa}} \widetilde{\mathbf{w}_a},\ \mathbf{w}_b^T \mathbf{S}_{Wb} \mathbf{w}_b \triangleq \widetilde{\mathbf{w}_b}^T \widetilde{\mathbf{S}_{Wb}} \widetilde{\mathbf{w}_b},$$
$$\mathbf{w}_a^T \mathbf{S}_{Ba} \mathbf{w}_a \triangleq \widetilde{\mathbf{w}_a}^T \widetilde{\mathbf{S}_{Ba}} \widetilde{\mathbf{w}_a},\ \ \mathbf{w}_b^T \mathbf{S}_{Bb} \mathbf{w}_b \triangleq \widetilde{\mathbf{w}_b}^T \widetilde{\mathbf{S}_{Bb}} \widetilde{\mathbf{w}_b}.$$

As a result of (3), the constraint conditions can be rewritten as:

$$\begin{pmatrix}\widetilde{\mathbf{w}_a}^T & \widetilde{\mathbf{w}_b}^T\end{pmatrix}\begin{pmatrix}\widetilde{\mathbf{S}_{Wa}} - k\widetilde{\mathbf{S}_{Ba}} & \mathbf{0}_{(d_a+1)\times(d_b+1)}\\ \mathbf{0}_{(d_b+1)\times(d_a+1)} & \mathbf{0}_{(d_b+1)\times(d_b+1)}\end{pmatrix}\begin{pmatrix}\widetilde{\mathbf{w}_a}\\ \widetilde{\mathbf{w}_b}\end{pmatrix} \le 0,$$

$$\begin{pmatrix}\widetilde{\mathbf{w}_a}^T & \widetilde{\mathbf{w}_b}^T\end{pmatrix}\begin{pmatrix}\mathbf{0}_{(d_a+1)\times(d_a+1)} & \mathbf{0}_{(d_a+1)\times(d_b+1)}\\ \mathbf{0}_{(d_b+1)\times(d_a+1)} & \widetilde{\mathbf{S}_{Wb}} - k\widetilde{\mathbf{S}_{Bb}}\end{pmatrix}\begin{pmatrix}\widetilde{\mathbf{w}_a}\\ \widetilde{\mathbf{w}_b}\end{pmatrix} \le 0.$$

### 2.3 Multi-view Fisher Discriminant Analysis

Suppose $v$ is the number of the views. The formulation of MFDA is:

$$\begin{aligned}\min\ \ & \sum_{i=1}^{n}\sum_{j=1}^{v}\alpha_j(y_i - \overset{\wedge}{y_{ji}})^2 + \sum_{i=1}^{n}\sum_{l=1}^{v}\sum_{t\neq l}^{v}\beta_{lt}(\overset{\wedge}{y_{li}} - \overset{\wedge}{y_{ti}})^2\\ \text{s.t.}\ \ & k\mathbf{w}_j^T \mathbf{S}_{Wj}\mathbf{w}_j - \ \mathbf{w}_j^T \mathbf{S}_{Bj}\mathbf{w}_j \le 0 \quad (j = 1, ..., v).\end{aligned} \tag{3}$$

Here we can see that MFDA still consists of three parts. The first part (the first term of the objective function) draws near each input's real label and its output of discriminant function. The second part (the second term of the objective function) makes views be consistent with others. Constant $\alpha_j$ and $\beta_{lt}$ control the relative importance of these competing parts. Constraint conditions (the third part) control the low range of the Fisher criterion value.

### 2.4 Classification for MFDA

Before using MFDA, the label $y_i$ is represented by a binary variable $\{-1, 1\}$. The predicted label is obtained by checking the symbols of outputs from the discriminant functions. Each view gives a predicted label, so that input $x_i$ has $v$ predicted labels. The final predicted label of input $x_i$ is determined by the view that has the maximum confidence. The confidence is given by $confidence_{ji} = abs(\overset{\wedge}{y_{ji}})$ where function $abs(z)$ gives the absolute value of $z$.

## 3 Hierarchical MFDA

There are two forms of between-class scatter in traditional FDA. The form, which can be applied to the multi-class case, separates class i from others by enlarging the distance between the average of class i and the total average. Some classes can not be separated from others well in this way. Take the data set in Fig.1 as an example. Here we can see it is not easy to find a good subspace to separate class three from others. If we first separate class one, two, three from class four, five, six, then further separate class one, two from class three, samples in class three would be classified with a higher accuracy. Thus we propose HMFDA which combines hierarchical metric learning [19], which has been proposed in our previous work, with MFDA to improve the performance of MFDA for multi-class case.

MFDA can be taken as a special case of HMFDA. MFDA only solves two-class problem, while HMFDA solves not only two-class problems, but also multi-class problems. There are four steps in MFDA. Firstly, two clusters are built according to the

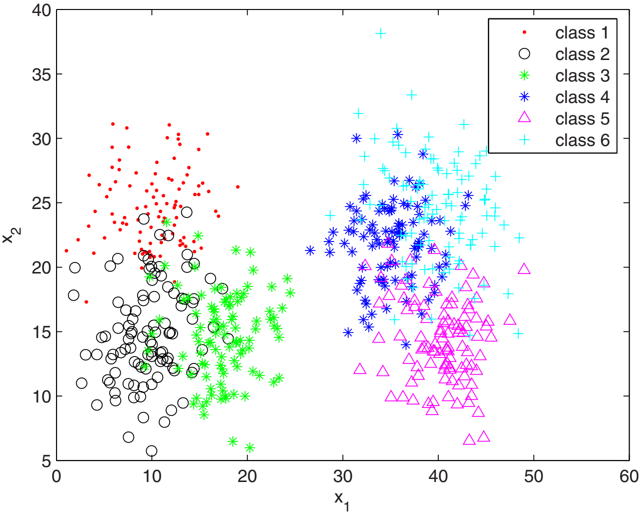


**Fig. 1.** The panel illustrates the distribution of six classes

clustering criterions. Secondly, optimal transforms to separate these two clusters are learned by MFDA. Thirdly, if a cluster includes more than one class, the first two steps are repeated until each cluster contains only one class. Finally, test points are classified hierarchically.

### 3.1 Clustering Criterion

Here we utilize two clustering criterions—overlapping ratio [19], which has been proposed in our previous work, and the minimum distance between clusters. The overlapping ratio is not sensitive to outliers and has no requirement for the shape and density of data distribution. Large overlapping ratio means big overlapping degree between clusters. If the overlapping ratio between two clusters is small, it is probably that two clusters can be easily separated. When two clusters can not be obtained just with the aid of overlapping ratios between classes, the minimum distance will be used.

### 3.2 Building Clusters

Clustering here consists of two steps. The first clustering step is guided by overlapping ratios between classes and the second step is guided by the minimum distance between clusters. But the second step is not necessary. If the first step returns only two clusters, the second step will be skipped over.

The procedure of clustering guided by overlapping ratio are shown as follows, where vector $\mathbf{A} = \{a_1, a_2, ..., a_p\}$ contains all nonzero overlapping ratios between classes in descending order:

```
– begin initialize $t \leftarrow 0$, $cluster_i \leftarrow class_i$ $(i = 1, 2, ..., c)$
    • do $t \leftarrow t + 1$
    • find the nearest classes according to the overlapping ratio $a_k$, say $class_i$ and $class_j$
    • find the clusters which $class_i$ and $class_j$ belongs to, say $cluster_i$ and $cluster_j$
    • merge $cluster_i$ and $cluster_j$, $c \leftarrow c - 1$
    • until $c = 2$ or $t = p$
– return the clusters
– end
```

### 3.3 Hierarchical Metric Learning for MFDA

As implied by the name, transformations here are learned by MFDA while in a hierarchical way. Each MFDA separates two clusters that obtained from clustering. The step is repeated until each cluster contains only one class.

For the example given in Fig. 1, MFDA needs to learn five transformations. Transformation $\mathbf{w}$ is learned to separate cluster one, which consists of class one, two, three, from cluster two which consists of class four, five, six. Then classes in cluster one and cluster two are grouped into sub-cluters. Transformation $\mathbf{w}_1$ is learned to separate class two, three from class one and transformation $\mathbf{w}_2$ is learned to separate class four, six from class five. The step is repeated until each cluster contains only one class.

### 3.4 Classification for Hierarchical MFDA

The test data are also classified hierarchically. Inputs are mapped into a feature space by the transformation. In this space, the test data are classified into a cluster with its final predicted label given by MFDA. This step will be repeated until the input is classified into a cluster which contains only one class.

Take Fig. 2 as an example. Firstly, inputs are mapped into a feature space by transformation $\mathbf{w}$ and the test data are classified by its final predicted label. Then, inputs are re-mapped by transformation $\mathbf{w}_1$ or $\mathbf{w}_2$ and the test data are sub-classified. The classification step is repeated until the test data are classified into a class.

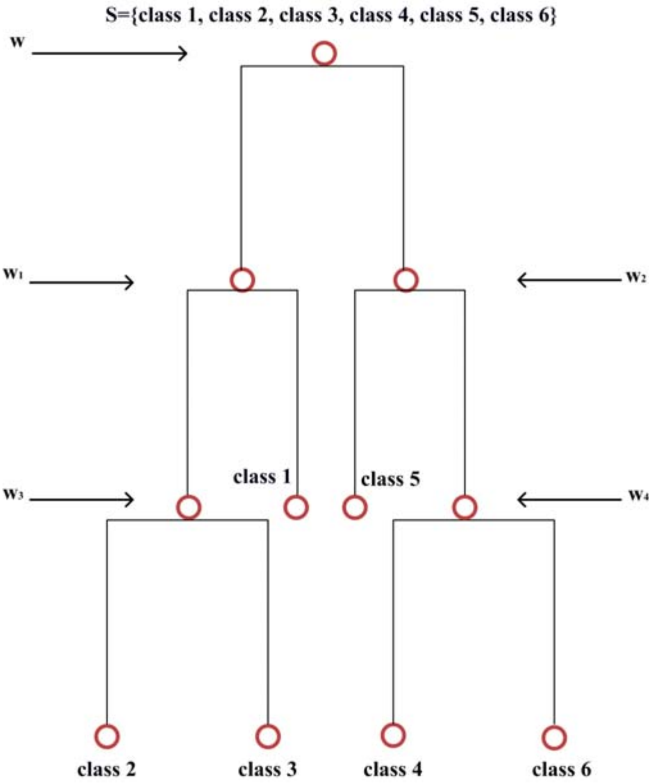


**Fig. 2.** The procedure of classifying a test data point for the given example

## 4 Experiments

### 4.1 Comparative Method

For the sake of objectively assessing our proposed method, we compare it with a single-view FDA whose projection maximizes the Fisher discriminant criterion value and the bias is sought by minimizing the distance between each point's output of the discriminant function and its real label. The objective function is $\min_{b} \sum_{i} (y_i - \overset{\wedge}{y_i})^2$. In this problem, hierarchical metric learning is also introduced to this method for the multi-class case.

### 4.2 Parameter Configuration

There are three parameters ($\alpha$, $\beta$, $k$) that need to be set. In our experiment, parameter $\alpha$ is set to be 1. While $\beta$, which ranges from 0.1 to 2.0 with the step 0.1, is chosen according to the training accuracies. The smallest $\beta$ is chosen when several ones have the same training accuracy will be chosen. The third parameter $k$ should be smaller than every view's Fisher criterion value.

### 4.3 Toy Data Sets

**Toy data creating and description.** Each view is composed of two parts. One part consists of some attributes of a Gaussian distribution. While the other part, which represents the correlation between views, is obtained through mixing some attributes from two different Gaussian distributions according to a certain proportion.

**Table 1.** Properties of Toy Data Sets

| | Class number | View dimension | Mixed dimension | Mixed proportion | Samples (training) | Samples (test) |
|---|---|---|---|---|---|---|
| Problem 1 | 2 | 2 | 1 | 0.3:0.7 | 60 | 40 |
| Problem 2 | 2 | 2 | 1 | 0.4:0.6 | 60 | 40 |
| Problem 3 | 2 | 4 | 1 | 0.2:0.8 | 60 | 40 |
| Problem 4 | 2 | 4 | 1 | 0.3:0.7 | 60 | 40 |
| Problem 5 | 2 | 5 | 2 | 0.2:0.8 | 60 | 40 |
| Problem 6 | 2 | 5 | 2 | 0.3:0.7 | 60 | 40 |
| Problem 7 | 3 | 5 | 2 | 0.2:0.8 | 60 | 40 |
| Problem 8 | 4 | 4 | 1 | 0.2:0.8 | 60 | 40 |
| Problem 9 | 4 | 4 | 2 | 0.2:0.8 | 60 | 40 |
| Problem 10 | 6 | 4 | 2 | 0.2:0.8 | 60 | 40 |

Take a two-view data set as an example. Suppose these two views are view a and view b and the dimensionality of each view is four. Two Gaussian distributions are created. Both of these two distributions are divided into two parts. Each part contains two attributes. Suppose $Apart1$, $Apart2$ are two parts for view a and $Bpart1$, $Bpart2$ are two parts for view b. The correlation between these two views are:

$$Corpart = Apart2 \times MixedPro1 + Bpart1 \times MixedPro2,$$

where $MixedPro1 + MixedPro2 = 1$. Two views are $\begin{pmatrix} Apart1 \\ Corpart \end{pmatrix}, \begin{pmatrix} Corpart \\ Bpart2 \end{pmatrix}$. In our toy data sets, there are ten problems whose properties are shown in Table 1.

**Classification results.** Table 2 shows the results for these ten problems. In these data sets, HMFDA outperforms the comparative method. It is not surprising that HMFDA works well since our toy data sets have more than one views, but the comparative method treats the problem by just a single view while our method addresses the problem from multiple views, therefore HMFDA performs better.

**Table 2.** Classification results (%) for problem 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

| | | Problem 1 | Problem 2 | Problem 3 | Problem 4 | Problem 5 |
|---|---|---|---|---|---|---|
| Training | FDA | 75.83 | 76.67 | 83.83 | 84.17 | 84.17 |
| accuracy | HMFDA | 79.17 | 80.00 | 85.00 | 84.17 | 85.00 |
| Test | FDA | 81.25 | 80.00 | 83.75 | 83.75 | 82.50 |
| accuracy | HMFDA | 83.75 | 82.50 | 85.00 | 83.75 | 85.00 |
| | | Problem 6 | Problem 7 | Problem 8 | Problem 9 | Problem 10 |
| Training | FDA | 83.33 | 89.44 | 52.08 | 40.42 | 61.67 |
| accuracy | HMFDA | 85.00 | 90.56 | 83.33 | 80.00 | 81.94 |
| Test | FDA | 86.25 | 84.17 | 47.05 | 33.13 | 64.58 |
| accuracy | HMFDA | 88.75 | 85.83 | 80.63 | 76.25 | 80.42 |

### 4.4 Real Data Sets

**Webpage data.** This data set consists of 1051 samples (230 course pages and 821 non-course pages). It has two instinct views, one view consisting of the words occurring in the page and the other view consisting of the words appearing in the link pointing to that page. The dimensionality of the first view is reduced from 2333 to 174 by PCA [1]. The dimensionality of the other view is 87. The classification results are shown in Fig. 3. Although HMFDA does not have the same accuracy as single-view FDA in the training data set, but it outperforms single-view FDA in the test data set.

**Advertisement data.** This data set consists of 3279 samples (459 advertisements and 2820 non-advertisements). It also has two natural views. Both of these two views are reconstructed through dimensionality reduction by PCA. The dimensionality of the first view is reduced from 587 to 167, while the other one is reduced from 967 to 167. As shown in Fig. 4, HMFDA outperforms single-view FDA both in the training data set and the test data set.

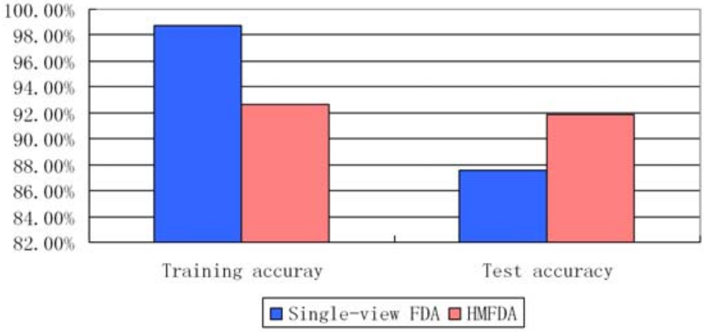


**Fig. 3.** Training and test accuracies for webpage data set

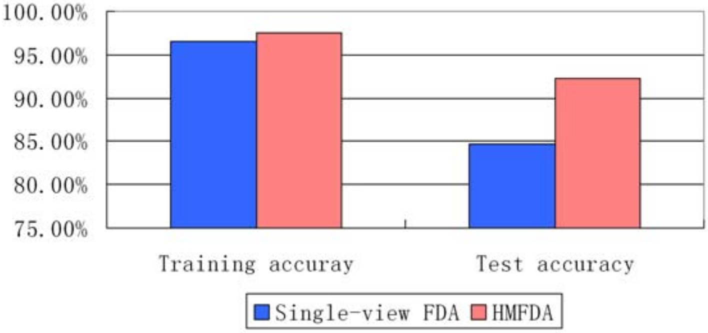


**Fig. 4.** Training and test accuracies for advertisement data set

## 5 Conclusion

Objects in the real world often have more than one natural feature set and therefore they can be described by more than one view. However, traditional FDA addresses problems by a single view. In this paper, we propose MFDA which combines traditional FDA with multi-view learning. Our approach is constructed by balancing those factors drawing near each input's real label and its predicted label, making views be consistent with others and enlarging the Fisher criterion value.

Sometimes, traditional FDA can not work well for multi-class data sets. It has been discussed in Section 4. Inspired by the idea of local metric learning, we further propose HMFDA which combines hierarchical metric learning [19] with MFDA.

In this paper, the transformation **w** has not been regularized. We will improve HMFDA by considering the regularization of **w** in the future.

## Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under Project 60703005, and by Shanghai Educational Development Foundation under Project 2007CG30.

## References

1. Duda, R.O., Hart, P.E., Stork, D.: Pattern Classification, 2nd edn. John Wiley and Sons, New York (2001)
2. He, Q.P., Qin, S.J., Wang, J.: A New Fault Diagnosis Method Using Fault Directions in Fisher Discriminant Analysis. AIChE Journal 51(2), 555–571 (2009)
3. Xu, Y., Yang, J., Jin, Z.: A Novel Method for Fisher Discriminant Analysis. Pattern Recognition 37(2), 381–384 (2004)
4. Duchene, J., Leclercq, S.: An Optimal Transformation for Discriminant and Principal Component Analysis. IEEE Transaction on Pattern Analysis and Machine Intelligence 10(6), 978–983 (1988)
5. Bo, L.F., Wang, L., Jiao, L.C.: Feature Scaling for Kernel Fisher Discriminant Analysis Using Leave-One-Out Cross Validation. Neural Computation 18(4), 961–978 (2006)
6. Kim, S.J., Magnani, A., Boyd, S.: Optimal Kernel Selection in Kernel Fisher Discriminant Analysis. ACM International Conference Proceeding Series 148, 465–472 (2006)
7. Sugiyama, M.: Dimensionality Reduction of Multimodal Labeled Data by Local Fisher Discriminant Analysis. The Journal of Machine Learning Research 8, 1027–1061 (2007)
8. Sun, S.: Semantic features for multi-view semi-supervised and active learning of text classification. In: Proceedings of the IEEE International Conference on Data Mining Workshops, pp. 731–735 (2008)
9. Sun, S.: High reliable multi-view semi-supervised learning with extremely sparse labeled data. In: Proceedings of 8th International Conference on Hybrid Intelligent Systems (HIS), pp. 935–938 (2008)
10. Diethe, T., Hardoon, D.R., Shawe-Taylor, J.: Multiview Fisher Discriminant Analysis. In: NIPS Workshop Learning from Multiple Sources (2008)
11. Hotelling, H.: Relations Between Two Sets of Variates. Biometrika 28, 312–377 (1936)
12. Sugiyama, M.: Local Fisher Discriminant Analysis for Supervised Dimensionality Reduction. In: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, Pennsylvania, pp. 905–912 (2006)
13. Domeniconi, C., Gunopulos, D., Peng, J.: Large Margin Nearest Neighbor Classifiers. IEEE Transactions on Neural Networks 16(4), 899–900 (2005)
14. Domeniconi, C., Peng, J., Gunopulos, D.: Locally Adaptive Metric Nearest-Neighbor Classificaton. IEEE Pattern Analysis and Machine Intelligence 24(9), 1281–1285 (2002)
15. Friedman, J.H.: Flexible Metric Nearest Neighbor Classification. Technical Report, Stanford University (1994)
16. Hastie, T., Tibshirani, R.: Discriminant Adaptive Nearest Neighbor Classification. IEEE Pattern Analysis and Machine Intelligence 18(6), 607–616 (1996)
17. Peng, J., Heisterkamp, D.R., Dai, H.K.: Adaptive Kernel Metric Nearest Neighbor Classificaton. In: Proceedings of 16th International Conference on Pattern Recognition, Quebec, Canada, pp. 33–36 (2002)
18. Yang, L., Jin, R., Sukthankar, R., Liu, Y.: An Efficient Algorithm for Local Distance Metric Learning. In: Proceedings of the National Conference on Artificial Intelligence (2006)
19. Chen, Q., Sun, S.: Hierarchical Distance Metric Learning for Large Margin Nearest Neighbor Classification. Submitted to International Journal of Pattern Recognition and Artificial Intelligence

# Auditory Temporal Assimilation: A Discriminant Analysis of Electrophysiological Evidence

Hiroshige Takeichi[1] , Takako Mitsudo[2], Yoshitaka Nakajima[3],
Gerard B. Remijn[4], Yoshinobu Goto[5], and Shozo Tobimatsu[2]

[1] Lab. for Mathematical Neuroscience, BSI, RIKEN,
2-1 Hirosawa, Wako-shi, Saitama 351-0198 Japan
{takeichi}@brain.riken.jp
[2] Dept. of Clin. Neurophys., Neurol. Inst., Faculty of Medicine, Kyushu University
[3] Dept. of Human Science, Faculty of Design, Kyushu University
[4] Grad. Sch. Human and Socio-Environment Studies, Kanazawa University
[5] Dept. of Occup. Therapy, Faculty of Rehabilitation, Int'l Univ. of Health and Welfare

**Abstract.** A portion of the data from an event-related potential (ERP) experiment [1] on auditory temporal assimilation [2, 3] was reanalyzed by constructing Gaussian Naïve Bayes Classifiers [4]. In auditory temporal assimilation, two neighboring physically-unequal time intervals marked by three successive tone bursts are illusorily perceived to have the same duration if the two time intervals satisfy a certain relationship. The classifiers could discriminate the subject's task, which was judgment of the equivalence between the two intervals, at an accuracy of 86–96% as well as their subjective judgments to the physically equivalent stimulus at an accuracy of 82–86% from individual ERP average waveforms. Chernoff information [5] provided more consistent interpretations compared with classification errors as to the selection of the component most strongly associated with the perceptual judgment. This may provide us with a simple but somewhat robust neurodecoding scheme.

**Keywords:** Discriminant analysis, Chernoff Information, Time perception, Neurodecoding.

## 1 Introduction

Electroencephalogram (EEG) recordings reflect the dynamics of information processing in the brain in terms of a time-varying electromagnetic field. The EEG is stochastic and requires a statistical description in order to understand the dynamics of interest. Conventional averaging is one of the simplest approaches, in which the mean of the potential with a probability distribution represents the statistics at each time point. In this average waveform, a component response localized in time is conventionally identified in relation to an event such as a stimulus. Event-related brain potentials (ERPs) as well as event-related synchronization (ERS) and desynchronization (ERD) are commonly used to investigate cognitive processes. For more distributed (e.g. multi-component) cognitive processes such as auditory temporal assimilation, a different approach is needed. Here we used Gaussian Naïve Bayes

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 299–308, 2009.

classifiers [4] and Chernoff information [5] in the statistical EEG data analysis to supplement and expand the original findings and to clarify the mechanisms of auditory temporal assimilation and time perception.

### 1.1 Auditory Temporal Assimilation: The Original Data

Nakajima and collaborators [2, 3] reported an illusion in time perception, which is now called auditory temporal assimilation. Two neighboring time intervals marked by three successive tone bursts seem to have the same duration if the first time interval (T1) and the second time interval (T2) satisfy the following relationship: -80ms ≤ T1-T2 ≤ 40ms. The term "auditory temporal assimilation" stems from the fact that physically different time intervals are perceived as equal, as if they *assimilate* each other. Mitsudo and collaborators [1] reported ERP experiments to clarify the mechanisms of auditory temporal assimilation and time perception. Neural mechanisms for time perception are not well understood, and a robust illusion generally reflects the nature or the constraints with which a biological system solves an ill-posed problem in perception.

Here we reanalyzed a portion of the data to examine the ERP results in more detail. The relevant experimental details are briefly reviewed here. ERPs were recorded from 19 electrodes on the scalp from 11 participants. Stimulus patterns consisted of two neighboring time intervals marked by three successive tone bursts (20ms, 1000Hz), in which the first time interval (T1) varied from 80 to 320 ms in seven steps and the second time interval (T2) was fixed at 200 ms. Participants pressed one of two buttons based on a judgment of whether or not T1 and T2 had the same duration in the judgment (J) condition, or arbitrarily in the no judgment (NJ) condition. Grand mean average waveforms were obtained for each of the J and NJ conditions for each stimulus. Electrophysiological data for the J condition showed a CNV-like ERP component in the frontal area during the T2 presentation, a P300-like component in the parietal area 370-500ms after the second tone burst, and a slow negative component (SNCt) in the right prefrontal area after the third tone burst. P300 and SNCt components were not present in the NJ condition. The results suggested that the P300 and CNV reflect attention and memory to T1 and that SNCt was linked to the decision. They additionally showed that SNCt activity in a 100ms window after the third marker was different between the stimuli with subjectively equal intervals and those with subjectively unequal intervals.

### 1.2 Illusion and Neurodecoding

A general strategy to find the neural correlate of an illusion or a subjective experience is to compare the brain states with different cognitive states from the same physical stimulus and condition, as is done with binocular rivalry or perception of ambiguous figures. It is also used to find a brain-derived signal for a subjective experience that is invariant across individual subjects or trials. Thus, it may be one of the purest opportunities for neurodecoding or mind-reading.

### 1.3 Goal of the Study

The purpose of the present study was to find a neurophysiological and informational correlate of auditory temporal assimilation that is invariant across individual subjects.

More specifically, we asked which component of the ERP contains the most information relevant to auditory temporal assimilation or subjective judgment. Note that it is possible for a signal component that is present when the subject is not performing the task to contain information about the judgment as well. Alternatively, a signal component that is only present when the subject processes a class of stimuli may not carry information about a specific judgment on the class or the stimulus.

## 2 Gaussian Naïve Bayes Classifiers and Chernoff Information

In order to evaluate how much information about the subjective judgment on the temporal intervals was contained in each of the EEG components, we constructed Gaussian Naïve Bayes (GNB) classifiers (see [4]) and evaluated their performance in terms of Chernoff information (CI) between the alternative classifiers, but not classification errors. Note that the GNB classifiers consist of likelihood functions that are probability distributions. While classification error depends on the discriminative power of the classifiers as well as the representative power of the samples, distance between the classifiers of the alternative classes reflects the discriminative power of the classifiers alone. This may be particularly relevant when the prior or marginal probabilities are difficult to estimate or different between the alternative classes or data, which is basically a sampling problem. This approach might also separate the problems with the discrimination from those with sampling, to help us identify the source of classification errors.

### 2.1 The Input Data

Mean average waveforms were computed for individual subjects for each combination of the J and NJ conditions and two of the T1 values: 120ms and 280ms. These were marginal cases in which different subjects showed different degrees of the illusion. On average across subjects, auditory temporal assimilation was observed in 85% and 25% of the trials, respectively. Similarly, mean average waveforms were computed separately for the same and different responses for individual subjects for the two T1 values.

The waveforms were then conventionally divided into three parts, each corresponding to a component reported in the ERP analysis in [1]. The CNV-like component was assumed to be reflected in the EEG recorded in a 400ms window after the second tone (hereafter the *early* interval) from a set of anterior electrodes (Fp1, Fp2, F7, F3, Fz, F4, F8, T3, C3, Cz, C4, T4). The P300-like component was assumed to be in a 400ms window after the third tone (hereafter the *late* interval) from the set of posterior electrodes (T5, P3, Pz, P4, T6, O1, O2). The SNCt component was assumed to be in the 400ms window after the 3rd tone from the anterior electrodes.

### 2.2 GNB Classifier

A general GNB classifier may be described as follows:

A GNB classifier computes a posterior probability for each data $x$ and classifies the data in the class with the maximum posterior probability. Namely:

$$class(x) = \arg\max_k \, p(c_k|x) \; . \tag{1}$$

$C_k$ is a class. According to the Bayes theorem, the posterior probability can be calculated from the likelihoods and the prior probabilities.

$$p(c_k|x) = \frac{p(c_k)p(x|c_k)}{p(x)} \; . \tag{2}$$

Here we first postulate that the likelihood function has a Gaussian distribution with a mean $\mu$ and a standard deviation $\sigma$ as defined in Equation 4 for each class $k$ at each time point $i$.

$$p(x_i|c_k) = \frac{1}{\sqrt{2\pi}\sigma_i^k} \exp\left( -\frac{(x_i - \mu_i^k)^2}{2(\sigma_i^k)^2} \right) \; . \tag{3}$$

$$\mu_i^k = \frac{1}{S}\sum_{j=1}^{S} x_i^j \quad \sigma_i^k = \frac{1}{S}\sum_{j=1}^{S} (x_i^j - \mu_i^k)^2 \; . \tag{4}$$

$S$ is the number of samples in each class. We further postulate that the prior probability for an unknown sample for each class is equal.

$$p(c_k) = const \; . \tag{5}$$

This assumption holds true for discrimination between the J and NJ conditions, but may not for discrimination between the same and different judgments. Additionally, we postulate the prior probability for observation of an unknown sample is equal.

$$p(x) = const \; . \tag{6}$$

Equation 6 may not be as intuitive as Equation 5. Equation 6 states that data $x$ at each time point $i$ are equally likely in their prior probability distribution, which is uniform. This means that under the assumption that $p(c_k)$ is constant,

$$\sum_k p(x_i \mid c_k) = const \; . \tag{7}$$

This may hold true as a rough approximation for data $\mu^1 < x < \mu^2$ when $\mu^1 < \mu^2$ and $\sigma^1 = \sigma^2$, but not otherwise. Finally, we postulate independence across time points, as shown in Equation 8.

$$p(x_i, x_j) = p(x_i)p(x_j) \; i \neq j \; . \tag{8}$$

Under these assumptions, a log-likelihood in Equation 9 may be used for classification. *N* is the number of time points.

$$p(c_k|x) = \frac{p(c_k)p(x|c_k)}{p(x)} \propto \log p(x|c_k) = \sum_{i=1}^{N} \log p(x_i|c_k) \; . \tag{9}$$

For each of the three components described in the previous section, a leave-one-out procedure was employed to construct classifiers to discriminate the waveforms between the J and NJ conditions, or the same and different judgments. Namely, a classifier was first constructed using data from ten out of eleven subjects, "leaving-one-out." Then the classifier was used to make a blind estimate as to which of the J and NJ or same and different classes was more likely for each of the two waveforms of the left-out subject. This procedure was repeated eleven times, varying the left-out subject, to obtain generalization performance.

### 2.3 Chernoff Information

In addition to the classification error, CI was computed between the two likelihood functions for the alternative classes in order to evaluate their discriminating power. CI can be defined as follows:

$$C(P_1, P_2) \equiv -\min_{0 \le \lambda \le 1} \log \left( \sum_{x} P_1^{\lambda}(x) P_2^{1-\lambda}(x) \right) \; . \tag{10}$$

It is the highest possible exponent for the probability of error in the Bayes case. We used CI in place of a more standard measure such as KL divergence, for it is symmetric and asymptotic.

## 3 Results

### 3.1 Discrimination between J and NJ: *Decoding the Task*

The discrimination accuracy and the CI for the anterior early CNV-like, the anterior late SNCt and the posterior late P300-like components for T1=120ms were 0.86 and 2.342, 0.86 and 2.598, and 0.86 and 2.155, respectively. The most discriminating component was SNCt in terms of Chernoff information. Similarly, those for T1=280ms were 0.86 and 2.645, 0.95 and 2.786, and 0.91 and 2.301. The most discriminating component was again SNCt (Fig. 1). This is in agreement with the previous ERP analysis that SNCt was most associated with temporal assimilation.

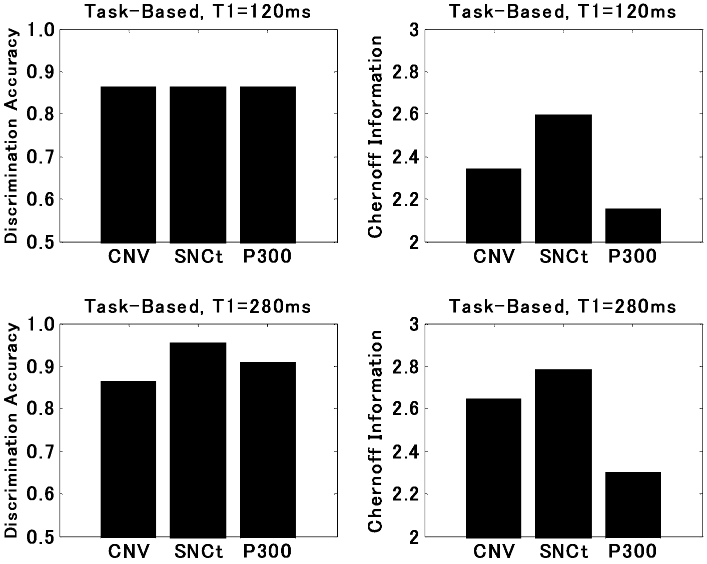


**Fig. 1.** Discrimination accuracy and Chernoff information for the task-based classification (N=11)

### 3.2 Discrimination between Same and Different: *Decoding the Judgment*

The discrimination accuracy and the CI for the anterior early CNV-like, the anterior late SNCt and the posterior late P300-like components for T1=120ms were 0.73 and 2.770, 0.59 and 2.925, and 0.82 and 2.801, respectively. The most discriminating component was SNCt in terms of CI and P300 in terms of discrimination error. Similarly, those for T1=280ms were 0.86 and 3.061, 0.82 and 3.209, and 0.73 and 2.994. The most discriminating component was SNCt in terms of CI and CNV in terms of discrimination error (Fig. 2).

### 3.3 Correlation between Chernoff Information and Individual Judgment

In addition, for judgment discrimination, there was a significant correlation (*r*=0.54, *p<0.1* and *r*=0.62, *p<0.05)* for CNV and SNCt but not P300 (*r*=0.44, *n.s.)* at T1=280ms between CI and the frequency of occurrence of temporal assimilation averaged across the subjects for the individual likelihood function (Fig. 3). No significant correlation was found when the likelihood function was for task discrimination (*r*=0.29, 0.46 and 0.27 for CNV, SNCt and P300, respectively, *n.s.*). This implies that the EEG waveforms, more specifically CNV and SNCt components may contain quantitative information about the individual subject's tendency on subjective time perception.

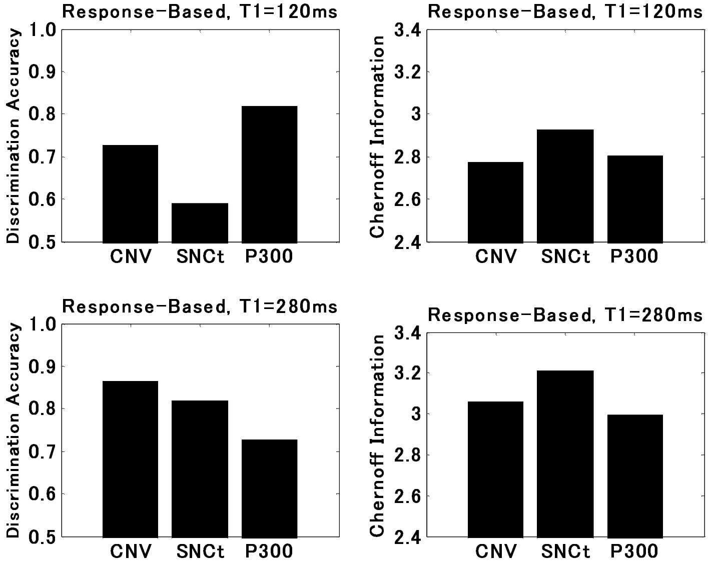


**Fig. 2.** Discrimination accuracy and Chernoff information for the response-based classification (N=11)

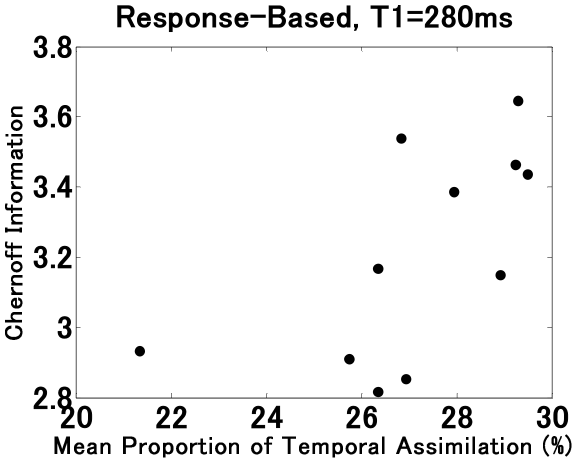


**Fig. 3.** Chernoff information between the likelihood functions for the two classes for individual subject groups (N=11), plotted against the rate of auditory temporal assimilation averaged across the corresponding subjects

### 3.4 Relation between Chernoff Information and KL Divergence

The primary purpose of the present study was to identify the most critical ERP component in predicting the occurrence of the illusion with discriminant analysis. The goodness of classification was evaluated by CI. It is known that the distance and discrimination error are related. Their discrepancy here may stem from non-Gaussian distribution of the sample. In this regard, we compared CI and KL divergence (KL). KL is more influenced than CI by the sample in the periphery of the distribution, where the probability density is close to zero. If the sample or the data from individual subjects exactly follows the assumed Gaussian distribution, these two measures should be consistent. Difference in distribution between the classes may have contributed to the discrepancy as well. For the task discrimination, the mean KL for CNV, SNCt and P300 at T1=120ms were 0.0491, 0.0629 and 0.0489, respectively. Those at T1=280ms were 0.1306, 0.1563 and 0.0851, respectively. For the judgment discrimination, the mean KL for CNV, SNCt and P300 at T1=120ms were 0.0366, 0.0192 and 0.0489, respectively. Those at T1=280ms were 0.0208, 0.0242 and 0.0280, respectively. KL showed a pattern similar to the discrimination accuracy.

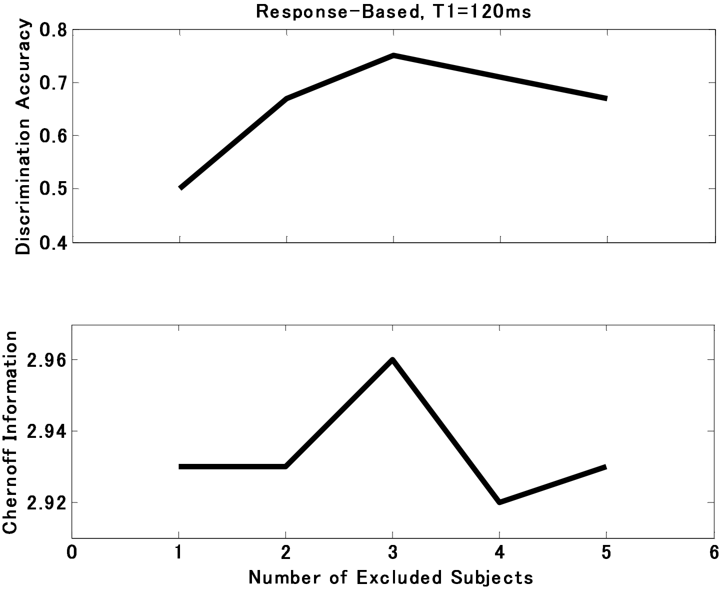


**Fig. 4.** Discrimination accuracy and Chernoff information are plotted against number of excluded subjects

For the judgment discrimination, CI and KL correlated more at T1=280 ms than at T1=120 ms. We assumed that the samples that introduced the discrepancy between CI and KL were 'outliers' and expected that resampling might counter their effects and reduce the discrimination error. We found that the discrimination accuracy for SNCt at T1=120ms was actually improved from 0.59 to 0.75 when three subjects were excluded from the analysis. The subjects were selected as follows. First, a scatter plot of the sample set was made between CI and KL. Then sample sets nearby the linear regression line were selected. Finally, the samples that had been left out in these sets were considered to be the outliers, for the CI and KL apparently showed a better agreement by excluding these subjects. The accuracy was best when the nearest three subjects were excluded as shown in Fig. 4.

On one hand, it could be interpreted that when the individual difference causes violation of an assumption, it increases the discrimination error. On the other hand, the violation may cause a discrepancy between different measures of distance or divergence between the classifier likelihood functions. These may imply that the sample may be pruned for a consistency across a family of distance measures between the likelihood functions. We have also shown that the discrimination error may be reduced by applying principal component analysis to the input data to satisfy the independence assumption in Equation 8 [6].

## 4 Conclusions

The GNB analysis here confirmed that SNCt was most strongly associated with auditory temporal assimilation, and it seems likely to be the neurophysiological and informational correlate of auditory temporal assimilation that is invariant across individual subjects. Three more points should be noted. First, average EEG waveforms corresponding to subjectively different perception for the physically identical stimulus could be decoded and discriminated with the best accuracy of 0.82 for T1=120ms and 0.86 for T1=280ms. Second, Chernoff information provided more consistent interpretations compared with the discrimination error as to the selection of the component most strongly associated with the perceptual judgment. Third, the discrimination error was reduced when the sample was pruned for consistency between Chernoff information and KL divergence. This may provide us with a simple but somewhat robust neurodecoding scheme, and a method for quantifying individual perceptual tendency based on EEG data.

**Acknowledgments.** This study was partially supported by KAKENHI 20530674 to HT, 20653054 to YN and 19390242 to ST. TM was supported by the JSPS Research Fellowships for Young Scientists. The authors are grateful to Drs. Kukjin Kang and Shun-Ichi Amari for suggestions and discussion.

## References

1. Mitsudo, T., Nakajima, Y., Remijn, G.B., Takeichi, H., Goto, Y., Tobimatsu, S.: Electrophysiological Evidence of Auditory Temporal Perception Related to the Assimilation Between Two Neighboring Time Intervals. NeuroQuantology 7, 114–127 (2009)
2. Nakajima, Y., ten Hoopen, G., van der Wilk, R.G.H.: A new illusion of time perception. Music Perception 8, 431–448 (1991)
3. Miyauchi, R., Nakajima, Y.: Bilateral assimilation of two neighboring empty time intervals. Music Perception 22, 411–424 (2005)

4. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2008)
5. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, Chichester (2006)
6. Inoue, Y., Ogawa, A., Arai, K., Matsumoto, H., Matsuzaki, N., Koyama, S., Toyomaki, A., Omori, T., Morotomi, T., Takeichi, H., Kitazaki, M.: Neural decoding from auditory event-related potentials: Comparison of statistical-classification methods and characteristics as an EEG analysis method. Japanese Journal of Psychonomic Science (in press, 2009)

# Web Snippet Clustering Based on Text Enrichment with Concept Hierarchy

Supakpong Jinarat[1], Choochart Haruechaiyasak[2], and Arnon Rungsawang[1]

[1] Massive Information & Knowledge Engineering
Department of Computer Engineering, Faculty of Engineering
Kasetsart University, Bangkok 10900, Thailand
g4885047@ku.ac.th, arnon@mikelab.net

[2] Human Language Technology Laboratory (HLT)
National Electronics and Computer Technology Center (NECTEC)
Thailand Science Park, Klong Luang, Pathumthani 12120, Thailand
choochart.haruechaiyasak@nectec.or.th

**Abstract.** Clustering web snippet results returned from search engine helps facilitate browsing and navigating for users. Due to the extremely short length of web snippets, many traditional clustering techniques which adopt the bag of words model often yields unsatisfactory clustering results. In this paper, we propose a method of text enrichment for improving performance of web snippet clustering. The main idea is to expand the original snippets with some related conceptual terms. We apply the Open Directory Project (ODP), a web taxonomy organized by humans, to provide the concept hierarchy of the web contents. Using a test data set of 240 queries, we performed the experiments by using two clustering techniques: K-means clustering as the non-overlapping approach and the Suffix Tree Clustering (STC) as the overlapping approach. Using the proposed text enrichment method, the K-means clustering yielded the overall performance improvement up to 15.51% based on the F1 measure. On the other hand, the Suffix Tree Clustering with text enrichment helped improve the performance up to 53.71%.

**Keywords:** web snippet clustering, concept hierarchy, text clustering, k-means clustering, suffix tree clustering, text enrichment.

## 1 Introduction

Presently, the number of web pages on the WWW has continuously increased with a rapid growth rate. Search engines are the most widely used tool for retrieving relevant web pages for the users. A search engine usually returns a large number of web snippets as the search results corresponding to a query from the user. Consequently, users must spend a lot of time browsing and navigating the search results for the most relevant information.

Many previous research works handled the problem of large number of search results by using text clustering technique. Text clustering is a method of organizing similar documents into the same group. The performance of text clustering

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 309–317, 2009.

is based on the criteria that the documents within the same cluster (intra-cluster documents) should be have higher relevancy between the documents in the same cluster than the documents outside the cluster (inter-cluster documents).

Some web snippet clustering methods use the bag of words representation to convert a document as a term-vector and compute the similarity between two documents [1], [3], [4] The work on web snippet clustering in [2], [6], [7] maintains the document as the phrases and construct a suffix tree from the phrases corresponding to a document. The suffix tree is used to construct the base clusters the set of documents that contain at least one common phrase. Many similarity metrics rely on the co-occurrence of terms between two documents.

However, in contrast to the traditional text clustering, the web snippet is a kind of short text document, which usually contains no more than thirty words. Because of its length, web snippet has not enough information for pattern matching. Traditional bag of words representation used for the text clustering methods is often unsatisfied. The previous work [5] solved the problem of less information of a short text by integrating some explicit conceptual terms found in the lexicographical knowledge like the Wordnet [8]. The limitation of Wordnet is it contains only conceptual terms of vocabularies found in lexical database for English. However, it does not contain modern name entity terms that appear often in a web snippet.

In this paper, we present an effective method to enrich web snippet document by integrating the concepts returned from concept hierarchy, the Open Directory Project [1] (ODP). The ODP is a web taxonomy organized by human as hierarchical structure of categories. We compare the clustering performance between a simple baseline and different source of features selection on text enrichment.

## 2 Text Enrichment with ODP Concept Hierarchy

The main idea of our work is to enrich the original text documents with some related conceptual terms, the documents that has the same concepts will be gathered into the same group. The concepts are retrieved from the concept hierarchy.

### 2.1 The Characteristic of Concept Hierarchy from ODP

We apply the ODP, web taxonomy that has edited by humans, as the concept hierarchy. A characteristic of concept hierarchy is the upper concepts are more general than the lower one. Oppositely, the lower concepts are more specific than the upper one as shown in Fig. 1.

In this work, we assume the words of internal node as *concepts* and we treat the ODP document in the leaf node as a *document under the concept*.

[1] Open Directory Project: http://www.dmoz.org

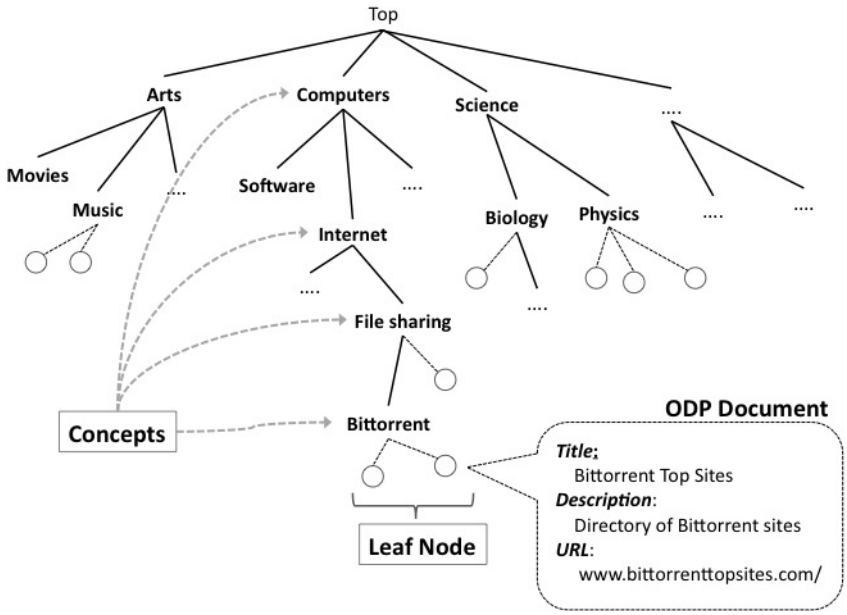


**Fig. 1.** The concept hierarchy from ODP

## 2.2 Our Proposed Method

We propose the methodology to enrich the document by extracting the concepts from the concept hierarchy and integrating concept to the document, as follows:

1. Select term features from the document, we define the features as words or terms, exclude the stop-words (such as a, the, foretc.).
2. Mapping the features to the leaf nodes in concept hierarchy and get all the concepts of the mapped leaf nodes return to be the candidate concepts
3. Apply top-N returned candidate concepts to construct a Suffix Tree. Apply from partial of suffix tree clustering algorithm (STC) in [2] use only 3 step:
   (a) Suffix tree construction step
   (b) Generating base clusters step
   (c) Merging base clusters step
4. Scoring the base clusters that are returned from step 2, based on document frequency in the base cluster, and then select only top score of base cluster.
5. Return a phrase which appears in the top base cluster as the concepts of the document. The concepts are integrated directly to the original document

In the step 2, the features will be mapped to the entire ODP documents at the leaf nodes in concept hierarchy. To speed up the mapping, we index all document at the leaf nodes with inverted index method using Lucene[2]The ODP

[2] The open source of Java API for indexing and searching technology provide by Apache Software Foundation, http://lucene.apache.org

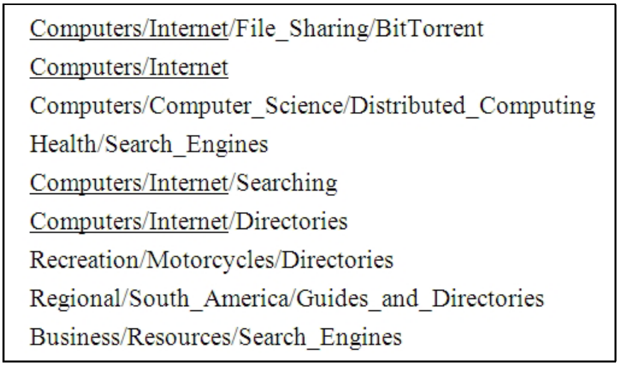
documents (leaf node) which are mapped with the features will be extracted all upper concepts of them as the candidate concepts. In step 3, we select top-N candidate concepts (number of N is set to 10 to 20 for speed up) to construct the suffix tree with the STC algorithm. In the base clusters merging of STC in this step, merging of any pair of overlapping base clusters are calculated by similarity measurement, $Sim\,(B_1, B_2)$, for two base cluster $B_1$ and $B_2$ with size, $|B_1|$ and $|B_2|$, respectively. If the similarity measure $Sim\,(B_1, B_2)$ equals to 1, the base clusters $B_1$ and $B_2$ will be merged. The function of $Sim\,(B_1, B_2)$ is defined as follows:

$$Sim(B_1, B_2) = \begin{cases} 1 & \text{if } \frac{|B_1 \cap B_2|}{|B_1|} > \alpha \wedge \frac{|B_1 \cap B_2|}{|B_2|} > \alpha \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

Where $\alpha$ is a merge threshold that has the range value between 0 and 1. In our experiment, we found that $\alpha$ is equal to 0.6 yields the best performance.

Steps 4, the base clusters from suffix tree are scored based on the length of phrase (the number of word in the base cluster) and the number of documents contained in the base cluster. We consider only the phrase of top score base cluster as the concepts. In the final step, the concepts are integrated directly into the original document.

For example, submission the term features from snippet: The largest BitTorrent search engine and directory on the net to the concept hierarchy and then returns the candidate concepts as shown in Fig. 2.



Computers/Internet/File_Sharing/BitTorrent
Computers/Internet
Computers/Computer_Science/Distributed_Computing
Health/Search_Engines
Computers/Internet/Searching
Computers/Internet/Directories
Recreation/Motorcycles/Directories
Regional/South_America/Guides_and_Directories
Business/Resources/Search_Engines

**Fig. 2.** Top ten candidate concepts returned from mapping the term features from the snippet: "*The largest BitTorrent search engine and directory on the net*"

The candidate concepts are used to construct a suffix tree, the method of merging base cluster and scoring are performed. The final base cluster in the suffix tree that has highest score is 'Computers Internet' that will be integrated to the original feature as "*The largest BitTorrent search engine and directory on the net. Computers Internet*". All of input features will pass into the step of text enrichment and then pass to the document clustering for the last step.

# 3 Experimental Setup

## 3.1 Data Collection

We classify the data collections into six groups. Each group has a group name as a topic. We collected the web snippet documents from query submission to Google[3] with 40 queries corresponding to each topic. We get the top 40 of web snippet results returned by Google that matched with a submitted query. Each group contains 1,600 web snippet documents and the data collection contains 9,600 web snippet documents as shown in Table 1.

**Table 1.** Group of Web Snippets returned from searching with Google

| Topic | Number of query | Number of web snippet/query | Number of web snippet/topic |
|---|---|---|---|
| Computers | 40 | 40 | 1,600 |
| Health | 40 | 40 | 1,600 |
| Arts_Entertainments | 40 | 40 | 1,600 |
| Home_Garden | 40 | 40 | 1,600 |
| Shopping | 40 | 40 | 1,600 |
| Sports | 40 | 40 | 1,600 |

## 3.2 Evaluation Metrics

We evaluate the clustering performance based on the standard measures of precision, recall and $F1$. Each cluster from the algorithm is labeled with the topic which has the largest number of documents. Let $C$ be the set of clusters from the algorithm, and $T$ be the set of topics from the ideal clustering. $C_t \in C$ is the set of clusters that are labeled with a particular topic $t \in T$

The precision $P(c)$ of cluster $c$ is defined as:

$$P(c) = \frac{\max_{t \in T} \{|D_{c,t}|\}}{|D_c|} \tag{2}$$

The recall $R(t)$ of a topic $t \in T$ is defined as:

$$R(t) = \frac{\bigcup_{c \in C} D_{c,t}}{|D_t|} \tag{3}$$

Where $D_c$ is the set of documents in cluster $c$, $D_t$ is the set of documents are labeled with topic $t$, and $D_{c,t}$ is the set of documents in cluster $c$ are labeled with topic $t$

The overall of precision and recall of clusters are the average of the individual cluster precisions and topic recalls weighted by the size of each cluster and each topic respectively:

[3] Google Search Engine: http://www.google.com

$$P = \frac{\sum_{c \in C} P(c)|D_c|}{\sum_{c \in C} |D_c|} \tag{4}$$

$$R = \frac{\sum_{t \in T} R(t)|D_t|}{\sum_{t \in T} |D_t|} \tag{5}$$

The standard $F1$ is the harmonic mean of precision and recall used as the overall measure of the performance

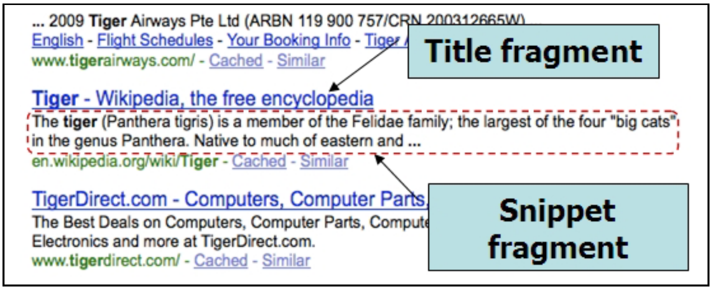
$$F1 = \frac{2PR}{P+R} \tag{6}$$

### 3.3 Feature Selection for Text Enrichment

We divided feature selection for web snippet (see Fig. 3) into three:

1. Enriching Text with Title fragment *(Enrich-Title)*: we use the features from Title fragment (contains 10 words on average) of web snippet as the input features for text enrichment method.
2. Enriching Text with Snippet fragment *(Enrich-Snippet)*: we use the Snippet fragment (contains 30 words on average) of web snippet document as the input features for text enrichment method.
3. Enriching Text with Full Web snippet document *(Enrich-Full)*: we use all features (contains 40 words on average) of web snippet document as the input features for text enrichment.

We performed the web snippet clustering with three feature selection strategies compare with the baseline (non enriching text)



... 2009 **Tiger** Airways Pte Ltd (ARBN 119 900 757/CRN 200312665W)
English - Flight Schedules - Your Booking Info - Tiger
www.tigerairways.com/ - Cached - Similar

Tiger - Wikipedia, the free encyclopedia
The **tiger** (Panthera tigris) is a member of the Felidae family; the largest of the four "big cats" in the genus Panthera. Native to much of eastern and ...
en.wikipedia.org/wiki/**Tiger** - Cached - Similar

TigerDirect.com - Computers, Computer Parts,
The Best Deals on Computers, Computer Parts, Compute
Electronics and more at TigerDirect.com.
www.tigerdirect.com/ - Cached - Similar

**Fig. 3.** The web snippet document returned by Google

# 4 Experimental Results

The primary objective of our experiment is to show the effect of text enrichment to the web snippet clustering. We divided the document clustering into two approaches. The first approach is non-overlapping clustering in which a document can appear in only one cluster. We used well-known K-means clustering as the non-overlapping approach. Second, the overlapping clustering approach, the document clustering that yield any document can appears in one or more clusters. In this approach, we used suffix tree clustering algorithm (STC).

In Fig. 4, the performance of K-means clustering of all feature selection for text enrichment is slightly higher than the baseline. We found that to enrich text with full web snippet provided the best performance.

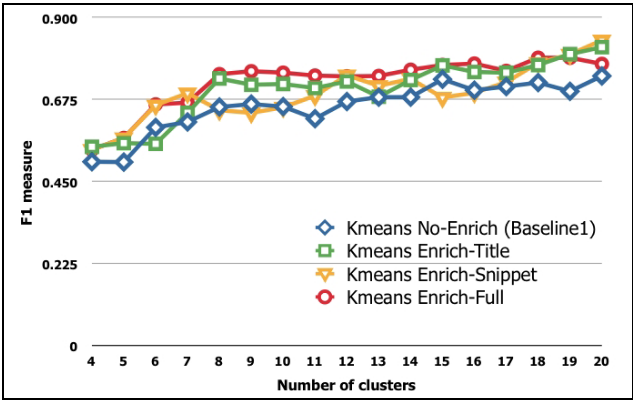


**Fig. 4.** Performance comparison of K-means clustering

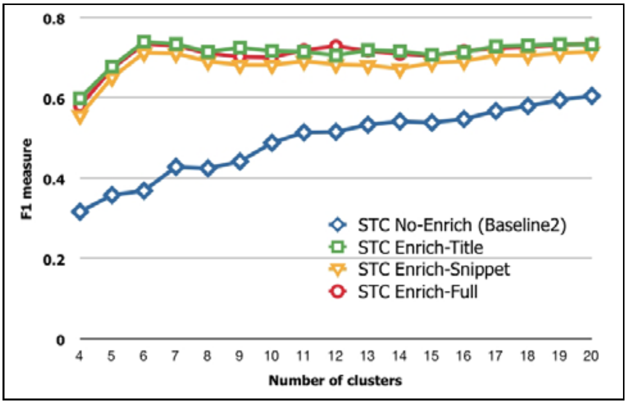


**Fig. 5.** Performance comparison of Suffix Tree clustering

In Fig. 5. The performance of STC, the overlapping clustering approach, with different feature selection strategies compared with the baseline. All text enrichment clustering achieve significantly better performance than the baseline. The best performance in this approach is presented in Enrich-Title feature selection strategy.

From the experiment results in Fig. 5 and Fig. 4, we calculated the average of performance by varying number of clusters of the both clustering approaches as shown in Table 2.

**Table 2.** The average performance of clustering compare with enriched text clustering and baseline. The values in parenthesis are percent of improvement.

| Method | Baseline (No enrich) | Enrich-Title | Enrich-Snippet | Enrich-Full |
|---|---|---|---|---|
| K-means | 0.679 | 0.779(+14.78%) | 0.752(+10.76%) | 0.784(+15.51%) |
| STC | 0.460 | 0.708(+53.71%) | 0.661(+43.60%) | 0.697(+51.36%) |

In Table 2, text enrichment effect to the performance of clustering obviously. Especially, the document clustering on the overlapping approach with STC algorithm. The returned concepts from the text enrichment method are the point of gathering for every document that contains the different words, but the same concepts.

## 5 Conclusion and Future Work

In this paper we have shown the achievement of web snippet clustering by enriching text with external knowledge from concept hierarchy like Open Directory Project. The idea of our approach is try to reduce the variance of document within one cluster by finding the common words (concepts) for join all documents in the same cluster. We tested the document clustering in the case of text enrichment with three feature selection. We found that some fragment of web snippet, such as Title, is enough to extract the appropriate concept that affect to the performance of clustering.

The main reason for using the ODP as background knowledge instance for the English lexicon to extract the concept of web snippet. Because, the ODP is a web taxonomy that contains a lot of information of web pages including technical terms or name entities and newly word these can not found in English lexicon database. In the future, we can improve the efficiency of extracting concept from Concept Hierarchy by finding the good taxonomy or expanding the knowledge of web taxonomy. We are attempting to apply our method of text enrichment to on-line web snippet clustering, and improving feature selection for extract the correct concept of web snippet.

**Acknowledgments.** This work has been supported by the National Electronics and Computer Technology Center (NECTEC) under grants TGIST-01-50-068.

## References

1. Ngo, C.L., Nguyen, H.S.: A method of web search result clustering based on rough sets. In: Proceeding of the 2005 IEEE/WIC/ACM, International Conference on Web Intelligence, pp. 673–679 (2005)
2. Zamir, O., Etzioni, O.: Web document clustering: A feasibility demonstration. In: Proceedings of SIGIR 1998, 21st ACM International Conference on Research and Development in Information Retrieval, Melbourne, AU, pp. 46–54 (1998)
3. Osinski, S.: Improving Quality of Search Results Clustering with Approximate Matrix Factorizations. In: Lalmas, M., MacFarlane, A., Rüger, S.M., Tombros, A., Tsikrika, T., Yavlinsky, A. (eds.) ECIR 2006. LNCS, vol. 3936, pp. 167–178. Springer, Heidelberg (2006)
4. Osinski, S., Stefanowski, J., Weiss, D.: Lingo: Search Result Clustering Algorithm Based on Singular Value Decomposition. In: Proceedings of International IIS: Intelligent Information Processing and Web Mining Conference, Zakopane, Poland, pp. 359–368 (2004)
5. Hotho, A., Staab, S., Stumme, G.: Wordnet improves Text Document Clustering. In: Proceeding of the SIGIR 2003 Semantic Web Workshop, pp. 541–544 (2003)
6. Crabtree, D., Gao, X., Andreae, P.: Improving Web Clustering by Cluster Selection. In: Proceeding of the 2005 IEEE/WIC/ACM, International Conference on Web Intelligence, pp. 172–178 (2005)
7. Zang, D., Dong, Y.: Semantic, Online, Hierarchical Clustering of Web Search Results. In: Yu, J.X., Lin, X., Lu, H., Zhang, Y. (eds.) APWeb 2004. LNCS, vol. 3007, pp. 69–78. Springer, Heidelberg (2004)
8. Miller, G.: Wordnet: A lexical database for English. CACM 38(11), 39–41 (1995)

# Maintaining Footprint-Based Retrieval for Case Deletion

Ning Lu, Jie Lu, and Guangquan Zhang

Decision Systems & e-Service Intelligence (DeSI) Lab
Centre for Quantum Computation & Intelligent Systems (QCIS)
Faculty of Engineering and Information Technology, University of Technology,
Sydney P.O. Box 123, Broadway, NSW 2007, Australia
{philiplu,jielu,zhangg}@it.uts.edu.au

**Abstract.** The effectiveness and efficiency of case-based reasoning (CBR) systems depend largely on the success of case-based retrieval. The case-base maintenance (CBM) issues become imperative and important especially for modern societies. This paper proposes a new competence model and a new maintenance procedure for the proposed competence model. Based on the proposed competence maintenance procedure, footprint-based retrieval (FBR), a competence-based case base retrieval method, is able to preserve its own retrieval effectiveness and efficiency.

**Keywords:** case-based reasoning, case-based maintenance, footprint-based retrieval, competence models.

## 1 Introduction

Case-based reasoning (CBR) systems solve new problems by retrieving and adapting solutions of previously solved problems that have stored in a case base. Obviously, the success of case-based problem solving is critically dependent on the retrieval of suitable cases [1]. Generally, a retrieval method is the combination of two procedures: a similarity assessment procedure to determine the similarity between a given case and target problem, and a procedure for searching the case memory in order to find the most similar cases [2]. Correspondingly, research on case base retrieval has two folds, with one fold focus on developing more efficient and accurate similarity assessment [1, 3, 4] and another focus on reducing the search needed to locate the best case without degrading competence or quality, such as footprint-based retrieval (FBR) [2], *k*d-tree [5], condensed nearest neighbor (CNN) [6]. Although these searching methodologies return suitable results and dramatically decrease the average searching time required, they tend to rely on some kinds of pre-structuring of the case memory.

Recently, the case base maintenance (CBM) issue has drawn more and more attention to the CBR community. The term CBM refers to the process of refining a CBR system to improve the system's performance, which implements policies of revising the contents of a CBR system in order to facilitate future reasoning [7]. For example, case deletion policy [8] or case addition policy [9, 10] is conducted to manage the case base knowledge in a CBR system. As a result of such changes in case memory, we also noticed a requirement for updating the indexing structures adopted in case

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 318–325, 2009.

base retrieval. Improperly managing of the indexing structure may degrade the retrieval efficiency or even fail to solve the target problem. For example, frequently changes in a case base may finally result an unbalanced tree, the deletion of cases in the reference set [6] may lead to unsuitable cases retrieved. Although the problem could be simply solved by restructuring with the whole case memory each time changes are made to the case base, it posts an unnecessary and undesirable cost to the CBR cycle, especially when changes are frequent and the restructuring cost is high. Thus it is imperative to develop new maintaining methods for case base retrieval.

In this paper, we suggest a new competence updating procedure for maintaining FBR [2]. Our work distinguishes from a previous work [11] in two aspects. First, we replace the *competence group* [12] with a new model - *competence closure*. We compare the differences between these two models, and show that our new competence model can still applied to the FBR. Second, we develop a new procedure for updating the competence model for case deletion which could be seen as a counterpart of Smyth and McKenna's work [11] which devoted to the scenario of case addition. Our work will contribute the research in the area of competence-based CBM.

This paper is organized as follow. Section 2 reviews the related works concerning the case base competence model [12] and FBR [2]. In Section 3, we first propose our new competence model, and then give out our algorithms of the proposed updating procedure. An example is given in Section 5 to illustrate the proposed updating procedure. Conclusions and future works come in Section 6.

## 2 Related Works

### 2.1 A Competence Model

According to Smyth and McKenna [12], the local competence of individual cases is characterized by its *coverage* and *reachability*. The *coverage* of a case is the set of target problems that it can be used to solve. The *reachability* of a target problem is the set of cases that can be used to provide a solution for the target. Since it is impossible to enumerate all possible future target problems, in practice the *coverage set* of a case is estimated by the set of cases that can be solved by its retrieval and adaption. And the *reachability set* of a case is estimated by the set of cases that can bring about its solution (see Definitions 1 and 2 respectively).

**Definition 1** [12]

$$\text{Given a case base } \mathrm{C} = \{\mathrm{c}_1, \dots, \mathrm{c}_n\}, \text{For } \mathrm{c} \in \mathrm{C},$$
$$\mathrm{CoverageSet(c)} = \{\mathrm{c'} \in \mathrm{C}: \mathrm{Solves(c, c')}\}$$

**Definition 2** [12]

$$\mathrm{ReachibilitySet(c)} = \{\mathrm{c'} \in \mathrm{C}: \mathrm{Solves(c', c)}\}$$

The *coverage set* and *reachability set* only provide a local estimation of competence. In order to estimate the true competence contributions of all cases in case base as a whole, the interactions between cases in terms of how their coverage and reachability sets overlap can further be modelled by *related set*, *shared coverage* and *competence group* [12]. The *related set* of a case is defined as the union of its *coverage set* and *reachability set* (Def. 3). The *shared coverage* of two cases exists if and only if the intersection of the *related sets* of two different cases is not empty (Def. 4). Finally all cases in a case base can be uniquely grouped together into *competence groups* (Def. 5) which are maximal sets of cases exhibiting shared coverage.

**Definition 3** [12]

$$\mathrm{RelatedSet}(c) = \mathrm{CoverageSet}(c) \cup \mathrm{ReachabilitySet}(c)$$

**Definition 4** [12]

$$\begin{aligned}&\text{For } c_1, c_2 \in C, \mathrm{SharedCoverage}(c_1, c_2)\\ &\qquad \text{iff } [\mathrm{RelatedSet}(c_1) \cap \mathrm{RelatedSet}(c_2)] \neq \emptyset\end{aligned}$$

**Definition 5** [12]

$$\begin{aligned}&\text{For } G = \{c_1 \cdots c_n\} \subseteq C, \quad \mathrm{CompentenceGroup}(G)\\ &\qquad \text{iff } \forall c_i \in G, \exists c_j \in G - \{c_i\}: \mathrm{SharedCoverage}(c_i, c_j) \wedge\\ &\qquad\qquad \forall c_k \in C - G, \nexists c_l \in G: \mathrm{SharedCoverage}(c_k, c_l)\end{aligned}$$

### 2.2 Footprint-Based Retrieval

Based on the competence model (Section 2.1), the *footprint set* of a case base is further defined as a minimal subset of the case base that covers all the cases in the case base with regarding to its competence [2]. It is constructed by computing *group footprint* for each *competence group* of a case-base.

Footprint-based retrieval (FBR) [2] is a case base retrieval method by using footprint cases and their *related sets*. The key idea of FBR is to dynamically find a subset of the original case base which closely related to the target problem, and search the best case within it. There are two separated stages required for FBR, with stage one, searching a reference case from the footprint set in order to identify a local region of the case-base that similar to the target problem and stage two, retrieving from the related set of the reference case to locate the best case for the target problem. Although FBR dramatically decreases its average retrieval time, while preserving high retrieval optimality, its success largely depends on a well established footprint set and a sound competence model of the case base. However, without proper methods to manage and maintain the footprint set and the competence model, the effectiveness and efficiency of FBR deteriorates.

## 3 A Complementary Procedure for Updating Competence Model

Similar to case addition [11], the deletion of a case in the case base should also trigger the update of the competence model. Thus to develop a procedure for updating competence model when deleting a case in the case base, we suggested a new competence model – *competence closure*. Based on this model, we developed an updating method for case deletion which contains three steps: update local competences, replace competence closure and setup footprint set.

### 3.1 A New Competence Model

Since a *competence group* (Def. 5) must contain at least two cases, this model may not be able to represent all cases inside a case base, when the case base contains some rare cases, or a case is removed from an existing *competence group* with only two cases. In addition, a case base maybe not able to be exclusively grouped by all competence groups it contained, which may cause problems when implementing this model in real world. In this sense, we suggest a new competence model - *competence closure* which is defined as the maximal set of cases linked together though their *related set* (Def. 6). We show the differences between these two models in Figure 1.

**Definition 6**

$$
\begin{gathered}
\text{For } \mathrm{G} = \{\mathrm{c}_1 \cdots \mathrm{c}_\mathrm{n}\} \subseteq \mathrm{C},\ \text{CompentenceClosure (G)} \\
\text{iff } \forall \mathrm{c}_\mathrm{i}, \mathrm{c}_\mathrm{j} \in \mathrm{G}, \text{if } \mathrm{c}_\mathrm{i} \neq \mathrm{c}_\mathrm{j}, \exists\{\mathrm{c}_{\mathrm{i}_1}, \mathrm{c}_{\mathrm{i}_2}, \cdots, \mathrm{c}_{\mathrm{i}_\mathrm{k}}\} \subseteq \mathrm{G}, \\
\text{st. SharedCoverage}\left(\mathrm{c}_{\mathrm{i}_\mathrm{j}}, \mathrm{c}_{\mathrm{i}_{\mathrm{j}+1}}\right) \neq \emptyset\ (\mathrm{j} = 0, \cdots, \mathrm{k}) \text{ where } \mathrm{c}_\mathrm{i} = \mathrm{c}_{\mathrm{i}_0}, \mathrm{c}_\mathrm{j} = \mathrm{c}_{\mathrm{i}_{\mathrm{k}+1}}\ \wedge \\
\forall \mathrm{c}_\mathrm{k} \in \mathrm{C} - \mathrm{G}, \nexists \mathrm{c}_\mathrm{l} \in \mathrm{G} \quad \text{st. SharedCoverage}(\mathrm{c}_\mathrm{k}, \mathrm{c}_\mathrm{l}) \neq \emptyset
\end{gathered}
$$

Based on our new competence model, the black cases in a case base C can be linked together with regarding to their shared coverage. Finally, all cases will be uniquely grouped into several *competence closures*. A *competence group* can be formed by the union of any *competence closures* with at least two cases. In this sense, a *competence closure* is also a *competence group*, if and only if it has at least two cases. But a *competence group* is not necessarily to be a *competence closure*. In addition, a rare case can setup its own *competence closure* ($CC_3$) if it has no similar cases, but this case cannot be modeled by the *competence group*. Clearly, our new competence model can be used to replace the *competence group* in FBR. By doing that, we are able to model all cases in a case base.

### 3.2 An Updating Algorithm

The competence model of a case base will be affected differently when different cases are removed. Based on whether the deleted case is a footprint case or not and whether the deletion of a case will cause splitting of existing *competence closure*, there are four possible situations when deleting a case in case base. When deleting a non-footprint case and no change to the *competence closure* is required, we only update the local competence estimation of those cases in the related set of the removed case. Actually, this could be seen as the basic scenario, since the local competence will be updated for all the other situations as well. When deleting a non-footprint case and new *competence closures* are created, we reserve the current *footprint set*, while assigning each footprint case of the original *competence closure*, which the removed case belongs to, to the right *competence closure* which it belongs to. When deleting a footprint case and no change to the *competence closure* is required, we replace the *footprint set* of current *competence closure,* which the removed case belongs to. When deleting a footprint case and new *competence closures* are created, we reconstruct both the *competence closures* and their corresponding *footprint sets*.

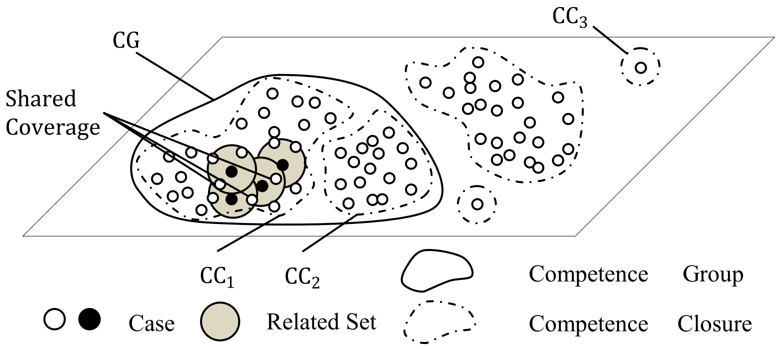


**Fig. 1.** Procedure for Update Local Competences

### 3.2.1 Update Local Competences

Different from the scenario of case addition, updating the local competence models of the remaining cases in the case base does not necessarily require checking the whole case base, but only the related ones. According to Def. 1 and Def. 2, we noted that for any case - *c*, a case - *x* belongs to the *coverage set* of *c*, if and only if *c* belongs to the *reachability set* of *x* (Theorem 1). In this sense, only the local competence of the cases in the *related set* of the deleted case will be affected. Thus we give out the update procedure of local competence for deletion (Algorithm. 1).

**Theorem 1**

$$\text{Given } \forall c \in C = \{c_1, c_2, \ldots, c_n\},\ \text{then we have } x \in \text{CoverageSet}(c)$$
$$\text{iff } c\ \in \text{ReachibiltySet}(x)$$

```
c ← deleted case
For each case x ∈ CoverageSet(c)
      Remove c from ReachabilitySet(x)
EndFor
For each case x ∈ ReachabilitySet(c)
      Remove c from CoverageSet(x)
EndFor
```

**Algorithm 1.** Updating Local Competences for Case Deletion

### 3.2.2 Replace Competence Closure

As the competence closure also changes along with the deletion of cases. Current competence closure containing the deleted case should be updated or replaced when new competence closures are created. We check whether the current competence closure still holds while setting up new groups when it does not (Algorithm. 2).

```
G ← competence closure, that c ∈ G
Remove c from G
replace-competence-Closure(G) {
      Create a case set m
      Get a case x from G
      Create a new competence closure G′ = G - {x}
      Add x into m
      while (m is not ∅)
            Get a case y from m
            For each case z ∈ RelatedSet(y)
                  If z ∈ G′
                        Add z into m
                        Remove z from G′
                  EndIf
            EndFor
            Remove y from m
      EndWhile
      If G′ is not {}
            Add G-G′ as a new competence closure
            Set G = G′
            call replace_competence_Closure(G)
      EndIf
}
```

**Algorithm 2.** Replacing Competence Closure for Case Deletion

### 3.3.3 Setup Footprint Set

Since the deletion of a non-footprint case will only affect the competence model behind the footprint set, but does not degrade the coverage of current footprint set. We assign each *competence closure* with its corresponding footprint set, but do not actually change the footprint set with regarding to the whole case base. However, if a case was deleted from the footprint set, we setup new footprint set for each *competence closure* built in Step 2 (Section 3.2.2). The algorithm of constructing footprint sets is given in Algorithm 3.

## 4 A Case Study

The deletion of cases from the case base will trigger the process of updating the competence model, which in turn may cause the replacement of current competence closure and selected footprint cases. In this section, we illustrate a simple scenario of deleting a non-footprint case, which finally results to the splitting of a competence closure, but remaining the footprint set unchanged.

As shown in Figure 2, cases in the case base have been grouped into two *competence closures*, and footprint cases have been chosen to facilitate case-base retrieval. When a non-footprint case is removed from the case base, the local competence model will be first updated. In our example, removing the case will lead to cases in a *competence closure* no longer linked all together through their shared coverage. Thus new *competence closures* will be created to replace the original one. Correspondingly, the selected footprint cases will be assigned to the new *competence closures* which they actually belong to.

```
CC ← competence closures used to replace G
FP ← FootprintSet(G)
If c ∈ FP
      For each competence closure g ∈ CC
            Setup footprint set for g
      EndFor
Else
      For each case x ∈ FP
            Find competence closure g∈CC, that x∈g
            Remove x from FootprintSet(G)
            Add x to FootprintSet(g)
      EndFor
EndIf
```

**Algorithm 3.** Setting up Footprint Set for Case Deletion

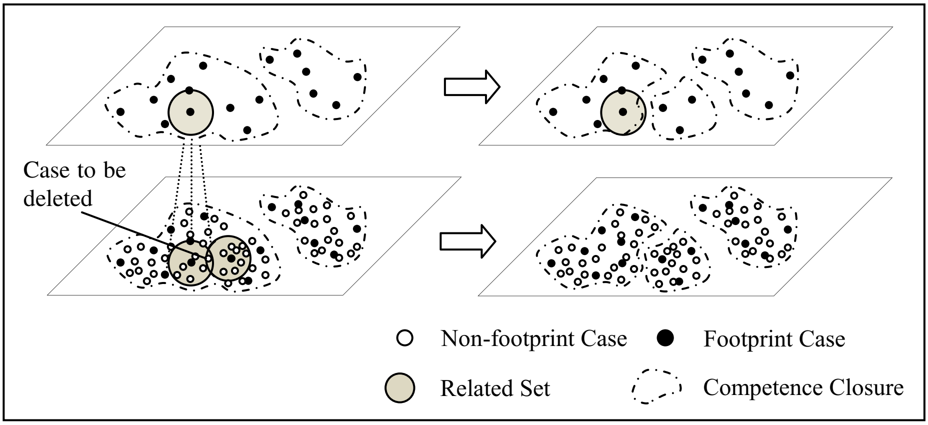


**Fig. 2.** Example of Deleting a Non-footprint Case

## 5 Conclusions and Future Works

Modern case base maintenance issues post challenges to the traditional case base retrieval methods. A certain case base retrieval method degrades its performance without proper maintenance. In this paper, we have proposed a new competence model - *competence closure*, and a new maintenance procedure for case deletion based on this new model. With our new model and the proposed procedure, FBR is able to maintain the competence model on which it based by itself, thus preserve its retrieval effectiveness and efficiency.

For future work, characteristics and relationships between our new competence model and Smyth and McKenna's models [12] still need further investigation. Also how to maintain a case base retrieval method when other forms of knowledge in a case base change will be another task.

**Acknowledgment.** The work presented in this paper was supported by Australian Research Council (ARC) under Discovery Project DP0880739.

## References

1. Smyth, B., Keane, M.T.: Retrieving Adaptable Cases: The Role of Adaptation Knowledge in Case Retrieval. In: Wess, S., Richter, M., Althoff, K.-D. (eds.) EWCBR 1993. LNCS, vol. 837, pp. 209–220. Springer, Heidelberg (1994)
2. Smyth, B., Elizabeth, M.: Footprint-Based Retrieval. In: Proceedings of the Third International Conference on Case-Based Reasoning and Development. Springer, Heidelberg (1999)
3. Castro, J.L., et al.: Loss and gain functions for CBR retrieval. Information Sciences 179(11), 1738–1750 (2009)
4. Cheng, W., Hüllermeier, E.: Learning Similarity Functions from Qualitative Feedback. In: Advances in Case-Based Reasoning, pp. 120–134 (2008)
5. Wess, S., Althoff, K.-D., Derwand, G.: Using k-d trees to improve the retrieval step in case-based reasoning. In: Topics in Case-Based Reasoning, pp. 167–181 (1994)
6. Hart, P.: The condensed nearest neighbor rule (Corresp.). IEEE Transactions on Information Theory 14(3), 515–516 (1968)
7. Wilson, D.C., Leake, D.B.: Maintaining Case-Based Reasoners: Dimensions and Directions. Computational Intelligence 17(2), 196 (2001)
8. Smyth, B., Keane, M.T.: Remembering To Forget: A Competence-Preserving Case Deletion Policy for Case-Based Reasoning Systems. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Francisco (1995)
9. Yang, Q., Zhu, J.: A Case-Addition Policy for Case-Base Maintenance. Computational Intelligence 17(2), 250 (2001)
10. Zhu, J., Yang, Q.: Remembering to Add: Competence-preserving Case-Addition Policies for Case-Base Maintenance. In: Proceedings of the International Joint Conference in Artificial Intelligence (IJCAI). Morgan Kaufmann, San Francisco (1999)
11. Smyth, B., McKenna, E.: An efficient and effective procedure for updating a competence model for case-based reasoners. In: Lopez de Mantaras, R., Plaza, E. (eds.) ECML 2000. LNCS (LNAI), vol. 1810, pp. 357–368. Springer, Heidelberg (2000)
12. Smyth, B., McKenna, E.: Competence Models and the Maintenance Problem. Computational Intelligence 17(2), 235 (2001)

# Investigation of Neonatal EEG Time Series Using a Modified Nonlinear Dynamical Analysis

Suparerk Janjarasjitt[1,3], Mark S. Scher[2], and Kenneth A. Loparo[3]

[1] Department of Electrical and Electronic Engineering, Ubon Ratchathani University,
Warinchamrab, Ubonratchathani 34190 Thailand
[2] Department of Neurology, Case Western Reserve University,
Cleveland, Ohio 44106 USA
[3] Department of Electrical Engineering and Computer Science,
Case Western Reserve University, Cleveland, Ohio 44106 USA
ensupajt@ubu.ac.th,
suparerk.janjarasjitt@case.edu,
mark.scher@uhhospitals.org,
kenneth.loparo@case.edu

**Abstract.** The Grassberger-Procaccia algorithm for computation of the correlation dimension is widely used nonlinear dynamical analysis techniques for EEG time series analysis. Even though the correlation dimension $D_2$ is the easiest dimension to compute, major drawback of the Grassberger-Procaccia algorithm is its extensive computational requirements. To overcome this, we introduce a modified computational algorithm referred to as the partial correlation integral. The partial correlation integral algorithm provides an approximation of the correlation dimension referred to as the dimensional exponent. Similar to the correlation dimension, the dimensional exponent can serve as a relative index of the complexity of a nonlinear dynamical system. In this study, the partial correlation integral algorithm is applied to analyze neonatal EEG sleep data. From the computational results, conclusions consistent with those made in previous studies using the correlation dimension are obtained.

**Keywords:** Neonate; EEG sleep; Nonlinear dynamics; Grassberger-Procaccia algorithm; Correlation dimension.

## 1 Introduction

Nonlinear dynamical analysis techniques derived from the theory of nonlinear dynamical systems including the correlation integral [1], Lyapunov exponents, approximate and sample entropy, etc. have been applied in a number of fields of research. In particular, some of these techniques have been proposed for the analysis of the human EEG time series [2]. In this context, nonlinear dynamical analysis methods have been widely applied to the study of both normal and abnormal EEG time series [3]. The spontaneous EEG time series reflects the functional states of neural networks in the brain [3] and nonlinear dynamical

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 326–335, 2009.

analysis of this signal can provide complementary information to improve our understanding of the dynamics of the brain in different functional (or physiological) states such as sleep, wake, rest, or pathology as compared with classical linear time series analysis methods such as Fourier or spectral analysis.

Among the available methods of nonlinear dynamical analysis, the correlation integral $C(r)$ using the Grassberger-Procaccia algorithm is the most common method that has been applied to the EEG time series [2], especially during sleep [4]. The correlation dimension $D_2$ computed from the Grassberger-Procaccia algorithm is the easiest dimension to compute [2]. A modification to the Grassberger-Procaccia correlation integral algorithm (introducing a new parameter, the Theiler window) to correct for autocorrelation effects in the time series that can lead to a lower estimate of the correlation dimension was proposed [5],[6]. When the Theiler window parameter is equal to 1, the modified correlation integral algorithm [5],[6] is exactly the same as the Grassberger-Procaccia algorithm.

In spite of computational difficulties and some shortcomings of the algorithm, the correlation integral and dimension are useful for making relative comparisons of the complexity of brain dynamics among groups of data [2]. The correlation dimension specifies the complexity of a deterministic nonlinear dynamical system in terms of the active degrees of freedom of the system on an attractor. Although easy to compute, major drawback of the Grassberger-Procaccia algorithm is computational complexity. In this study, we introduce a modified computational algorithm for computation of the Grassberger-Procaccia correlation integral that is referred to as the partial correlation integral. The primary advantage of the partial correlation integral algorithm is its reduced computational complexity and time. The partial correlation integral provides a good approximation of the correlation integral, and the exponent characterizing the slope of the partial correlation integral according to a power law is an approximation of the correlation dimension $D_2$. This exponent is referred to as the dimensional exponent.

In this study, we examine the complexity of neonatal EEG sleep data using the partial correlation integral. The dimensional exponents (complexity) of the neonatal EEG sleep data will be investigated into two aspects: subject groups and sleep states. The dimensional exponents of two different subject groups (preterm and full-term) will be evaluated. Further, the dimensional exponents of two different sleep states (active sleep and quiet sleep) will be examined. From the computational experiments, the results show that the dimensional exponent of the neonatal EEG sleep data of the full-term subject tends to be higher than that of the preterm subject even at the same postmenstrual age (PMA). Furthermore, the dimensional exponent associated with the active sleep state tends to be higher than that associated with the quiet sleep state. This supports the results of previous studies using the correlation integral and correlation dimension reported in [7],[8].

# 2 Methods

## 2.1 Correlation Integral and Dimension

The estimation algorithm for computation of the correlation dimension $D_2$ can be divided into two basic steps. In the first step, the attractor of the nonlinear system is reconstructed from a univariate time series using time-delay embedding, also referred to as the Takens reconstruction [9]. In step 2, the correlation dimension of the attractor is estimated from the correlation integral. The commonly used method for estimating the correlation dimension, such as in the nonlinear analysis of EEG [2], is the Grassberger-Procaccia algorithm (GPA) [1].

**Attractor Reconstruction.** Given a univariate time series $\{x[0], x[1], \ldots, x[N-1]\}$ where each sample $x[n]$ is a 1-dimensional (observed) measurement of the nonlinear system that is being analyzed. To provide a more complete description of the higher dimensional nonlinear system, the time series $x$ needs to be unfolded into a higher dimension space called the embedding space using a time-delay embedding scheme as, for example, given in [10]. The $m$-dimensional embedding vector $\mathbf{x} \in \mathbb{R}^m$ of the time series $x$ is given by [6]

$$\mathbf{x}[n] = \left( x[n] \; x[n+\tau] \; \cdots \; x[n+(m-1)\tau] \right)^T \tag{1}$$

where $n = 0, 1, \ldots, N_e - 1$, $N_e = N - (m-1)\tau$, and $m$ and $\tau$ are the embedding parameters denoting the embedding dimension and the time-delay, respectively, and $\cdot^T$ denotes vector transpose. This time-delay embedding technique unfolds the observed time series into a $m$-dimensional embedding space and provides a more comprehensive representation of the behavior of the higher dimensional nonlinear system on the attractor. There are a number of methods for determining the time delay $\tau$ such as the autocorrelation function [11], mutual information [12], higher-order correlation [13], average displacement [14], etc. A sufficient embedding dimension $m$ can be determined by using the false nearest neighbor technique [15], for example.

**Correlation Integral and Dimension Calculation.** The correlation integral $C(r)$ of the nonlinear time series $x[n]$ is defined by [1]

$$C(r) = \lim_{N_e \to \infty} \frac{2}{N_c} \sum_{i=0}^{N_e-1} \sum_{j=i+1}^{N_e-1} \Theta\left(r - \|\mathbf{x}_i - \mathbf{x}_j\|\right) \tag{2}$$

where $N_c = N_e(N_e - 1)$ and the Heaviside function $\Theta(n) = 1$ if $n \geq 0$; 0 otherwise. A revised algorithm was introduced by Theiler [5],[6] to correct for autocorrelation effects in the time series by using the Theiler window $w$. Theiler's modified algorithm is defined by [5],[6]

$$C(r) = \lim_{N_e \to \infty} \frac{2}{N_c} \sum_{i=0}^{N_e-1} \sum_{j=i+w}^{N_e-1} \Theta\left(r - \|\mathbf{x}_i - \mathbf{x}_j\|\right) \tag{3}$$

where $N_c = (N_e - w)(N_e - w + 1)$ and $w$ denotes the Theiler window.

According to Grassberger and Procaccia [1], the correlation integral $C(r)$ behaves as a power law for small $r$, that is

$$C(r) \propto r^{\nu}. \tag{4}$$

The exponent $\nu$ is defined as the correlation dimension $D_2$ and can be calculated by

$$\nu = \lim_{r \to 0} \frac{\log(C(r))}{\log(r)}. \tag{5}$$

### 2.2 Partial Correlation Integral and Dimensional Exponent

The partial correlation integral is a modified computational algorithm for calculation of the Grassberger-Procaccia correlation integral. The correlation integral algorithm measures all possible pairwise distances of points on the attractor in the embedding space. In contrast to the correlation integral algorithm, the partial correlation integral algorithm measures only those pairwise distances of points within a specific time window that is defined by the integral limit parameter of the partial correlation integral.

The partial correlation integral of the nonlinear time series $x[n]$ is defined by

$$C_\rho(r) = \frac{2}{\rho N_\rho} \sum_{i=0}^{N_\rho - 1} \sum_{j=i+w}^{i+(w+\rho-1)} \Theta\left(r - \|\mathbf{x}_i - \mathbf{x}_j\|\right) \tag{6}$$

where $\rho$, the integral limit of the partial correlation integral, is an arbitrary positive integer and $N_\rho = N_e - (w + \rho - 1) - 1$. The partial correlation integral $C_\rho(r)$ is a good approximation of the correlation integral $C(r)$ and also exhibits a power law behavior for small $r$, i.e.,

$$C(r) \propto r^{\nu_\rho}. \tag{7}$$

The exponent $\nu_\rho$ is defined as the dimensional exponent and is an approximation to the correlation dimension $D_2$.

Obviously, the partial correlation integral algorithm has less computational complexity than the Grassberger-Procaccia correlation integral algorithm. The computational complexity of the partial correlation integral algorithm is linear in the length of the attractor $N_e$, i.e. $\mathcal{O}(N_e)$, while the computational complexity of the correlation integral algorithm is $\mathcal{O}(N_e^2)$ if the length of the attractor $N_e$ is large.

## 3 Analytic Framework

### 3.1 Data and Subjects

Electroencephalographic/polysomnographic studies were performed in an environmentally controlled setting. Fourteen channels of bipolar EEG recording consisting of Fp1-T3, T3-O1, Fp2-T4, T4-O2, Fp1-C3, C3-O1, Fp2-C4, C4-O2, T3-C3, C3-C7, Cz-C4, C4-T4, Fz-Cz and Cz-Pz were obtained using the standard 10-20 EEG lead system. The neonatal EEG sleep data were recorded with a 12-bit A/D converter and a sampling rate of 64 Hz.

In this study, the subjects were categorized into two subject groups according to the gestational age of the subjects: preterm (PT) and full-term (FT). The neonatal EEG sleep data examined in this study were obtained from two subjects (one for each subject group). The preterm subject is 29 weeks gestational age while the full-term subject is 42 weeks gestational age. The EEG data of both subjects were however obtained when the subjects were both at 42 weeks postmenstrual age (PMA).

Digitized neurophysiologic data for each minute of sleep during the recording were compared with the contemporaneous minute of EEG sleep, and neonates were visually assigned one of six sleep states according to conventional neonatal EEG sleep criteria [16], i.e., two active sleep (AS) and two quiet sleep (QS) states as well as indeterminate and waking states.

### 3.2 Computational Experiments

Only channel Fp1-C3 of the neonatal EEG sleep data is analyzed in this work. The dimensional exponent of one-minute epochs of the neonatal EEG time series is determined. In the computation of the partial correlation integral and the dimensional exponents, the parameters used are as follows: the embedding dimension $m = 14$, the time delay $\tau = 2$, the Theiler window $w = 1$ and the integral limit $\rho = 800$.

In this study, the dimensional exponents of the neonatal EEG sleep data were investigated in two aspects. In the first aspect, the dimensional exponents of the neonatal EEG sleep data of the full-term subjects are compared to that of the preterm subjects associated with the same sleep state. In another aspect, the dimensional exponents of the neonatal EEG sleep data associated with the active sleep states are compared to that associated with the quiet sleep states for both subjects.

## 4 Results

The dimensional exponents and the correlation dimensions of the neonatal EEG sleep data are summarized in Table 1 and Table 2, respectively.

### 4.1 Comparison between the PT and FT Subjects

First, the dimensional exponents of the neonatal EEG sleep data of the full-term and preterm subjects associated with the same sleep states are investigated. The dimensional exponents of the neonatal EEG sleep data associated with the active sleep state of both the preterm and full-term subjects are compared in Fig. 1a. Obviously, the dimensional exponent of the neonatal EEG sleep data associated

**Table 1.** Dimensional Exponents of the Neonatal EEG Sleep Data

| Subject Group | Sleep State | Mean of $\nu_\rho$ | Std. of $\nu_\rho$ |
|---|---|---|---|
| FT | AS | 5.3554 | 0.4147 |
| FT | QS | 4.7267 | 0.5113 |
| PT | AS | 4.9192 | 0.5752 |
| PT | QS | 4.6929 | 0.3933 |

**Table 2.** Correlation Dimensions of the Neonatal EEG Sleep Data

| Subject Group | Sleep State | Mean of $D_2$ | Std. of $D_2$ |
|---|---|---|---|
| FT | AS | 6.2190 | 0.4025 |
| FT | QS | 5.4796 | 0.4889 |
| PT | AS | 5.6108 | 0.4845 |
| PT | QS | 5.4397 | 0.4777 |

with the active sleep state of the full-term subject tends to be higher than that of the preterm subject. Further, from a two-tail, paired $t$-test, the result suggests that there are statistically significant differences in the dimensional exponents between these two full-term and preterm subjects during the active sleep state ($p \ll 0.05$). In addition, the correlation dimensions of the neonatal EEG sleep data associated with the active sleep state of both the preterm and full-term subjects are compared in Fig. 2a. Similarly, the correlation dimension of the neonatal EEG sleep data associated with the active sleep state of the full-term subject tends to be higher than that of the preterm subject. The result of a two-tail, paired $t$-test also suggests that there are statistically significant differences in the correlation dimensions between the full-term and preterm subjects during the active sleep state ($p \ll 0.05$).

Fig. 1b compares the dimensional exponents of the neonatal EEG sleep data associated with the quiet sleep state of both the preterm and full-term subjects. It is shown that during the quiet sleep state the dimensional exponent of the neonatal EEG sleep data of the preterm subject is not significantly different from that of the full-term subject. However, the mean value of the dimensional exponents corresponding to the full-term subject is higher than that corresponding to the preterm subject as shown in Table 1. The correlation dimensions of the neonatal EEG sleep data associated with the quiet sleep state of both the preterm and full-term subjects are illustrated in Fig. 2b. There are not statistically significant differences in the correlation dimensions between the full-term and preterm subjects during the quiet sleep state.

### 4.2 Comparison between Sleep States

The dimensional exponents of the neonatal EEG sleep data associated with active and quiet sleep states for the preterm and full-term subjects are investigated.

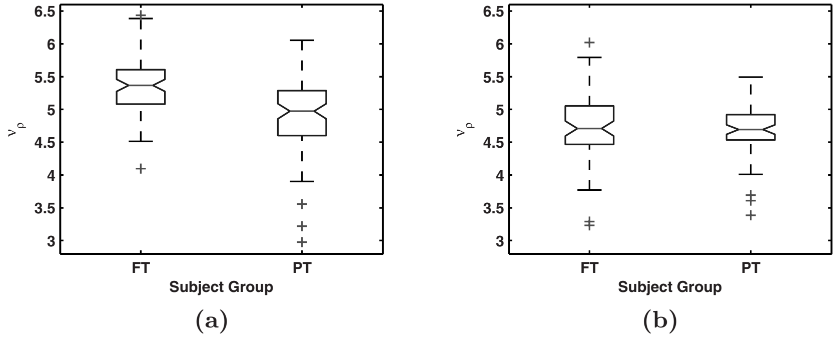


**Fig. 1.** Comparison of the dimensional exponents of the neonatal EEG sleep data associated with: (a) the active sleep state; and (b) the quiet sleep state

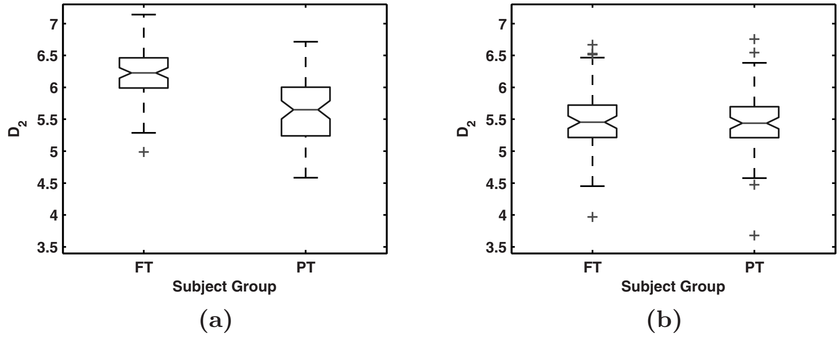
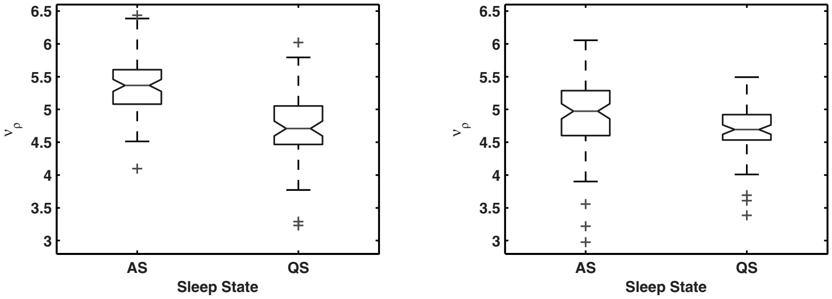


**Fig. 2.** Comparison of the correlation dimensions of the neonatal EEG sleep data associated with: (a) the active sleep state; and (b) the quiet sleep state

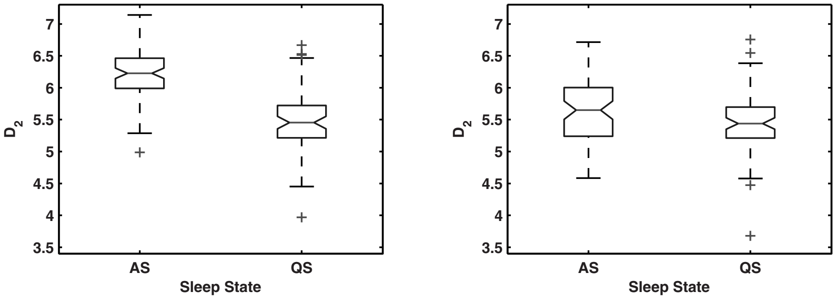
The dimensional exponents of the neonatal EEG sleep data for the full-term subject in the active and quiet sleep states are compared in Fig. 3a while Fig. 3b compares the dimensional exponents of the neonatal EEG sleep data for the preterm subject in the active and quiet sleep states. The dimensional exponent of the neonatal EEG sleep data during active sleep state tends to be higher than that associated with quiet sleep state for both the full-term and preterm subjects. Furthermore, the dimensional exponent of the neonatal EEG sleep data for the full-term subject in active sleep state is significantly higher than that associated with quiet sleep state ($p \ll 0.05$). The dimensional exponent of the neonatal EEG sleep data for the preterm subject during active sleep state is also significantly higher than during quiet sleep state ($p < 0.05$).

Fig. 4a illustrates the correlation dimensions of the neonatal EEG sleep data for the full-term subject in the active and quiet sleep states. The correlation



(a) (b)

**Fig. 3.** Comparison of the dimensional exponents of the neonatal EEG sleep data of: (a) the full-term subject; and (b) the preterm subject



(a) (b)

**Fig. 4.** Comparison of the correlation dimensions of the neonatal EEG sleep data of: (a) the full-term subject; and (b) the preterm subject

dimension of the neonatal EEG sleep data for the full-term subject in active sleep state is significantly higher than that associated with quiet sleep state ($p \ll 0.05$). The correlation dimensions of the neonatal EEG sleep data for the preterm subject in the active and quiet sleep states are compared in Fig. 4b. Correspondingly, the correlation dimension of the neonatal EEG sleep data for the full-term subject in active sleep state is significantly higher than that associated with quiet sleep state ($p < 0.05$).

## 5 Discussion

In this study, the modified computational algorithm referred to as the partial correlation integral algorithm is introduced. The partial correlation integral algorithm provides an approximation of the Grassberger-Procaccia correlation integral. Further, the dimensional exponent obtained from the partial correlation integral is an approximation of the correlation dimension $D_2$. The computational complexity and time of the partial correlation integral algorithm is much less than that of the Grassberger-Procaccia algorithm. The computational complexity of the partial correlation integral algorithm is linear in the length of the attractor $N_e$, i.e. $\mathcal{O}(N_e)$, while the computational complexity of the correlation integral algorithm is $\mathcal{O}(N_e^2)$ if the length of the attractor $N_e$ is large. The partial correlation integral algorithm was applied to machine vibration signal analysis in [17].

In this work, the partial correlation integral algorithm is applied to neonatal EEG sleep data analysis to investigate some aspects of neurodevelopment. From the computational results, it is observed that the neonatal EEG sleep time series of the full-term subject tends to be more complex than that of the preterm subject even at the same postmenstrual age. Also the computational results suggest that the neonatal EEG sleep time series during active sleep is more complex than during quiet sleep for both the full-term and preterm subjects. These results support the previous findings reported in [7],[8].

## Acknowledgment

Dr. Janjarasjitt is supported by a grant from Ubon Ratchathani University.

## References

1. Grassberger, P., Procaccia, I.: Characterization of strange attractors. Phys. Rev. Lett. 50, 346–349 (1983)
2. Pritchard, W.S., Duke, D.W.: Measuring chaos in the brain: a tutorial review of EEG dimension estimation. Brain Cogn. 27, 353–397 (1995)
3. Stam, C.J., Pritchard, W.S.: Dynamics underlying rhythmic and non-rhythmic variants of abnormal, waking delta activity. Int. J. Psychophysiol. 34, 5–20 (1999)
4. Ferri, R., et al.: Non-linear EEG measures during sleep: effects of the different sleep stages and cyclic alternating pattern. Int. J. Psycholphysiol. 43, 273–286 (2002)
5. Theiler, J.: Spurious dimension from correlation algorithms applied to limited time-series data. Phys. Rev. A. 34, 2427–2432 (1986)
6. Theiler, J.: Estimating fractal dimension. J. Opt. Soc. Am. A. 7, 1055–1073 (1990)
7. Janjarasjitt, S., Scher, M.S., Loparo, K.A.: Nonlinear dynamical analysis of the neonatal EEG time series: the relationship between neurodevelopment and complexity. Clin Neurophysiol. 119, 822–836 (2008)
8. Janjarasjitt, S., Scher, M.S., Loparo, K.A.: Nonlinear dynamical analysis of the neonatal EEG time series: the relationship between sleep state and complexity. Clin. Neurophysiol. 119, 1812–1823 (2008)
9. Takens, F.: Detecting strange attractors in turbulence. Dynamical Systems and Turbulence 898, 366–381 (1981)
10. Packard, N.H., Crutchfield, J.P., Farmer, J.D., Shaw, R.S.: Geometry from a time series. Phys. Rev. Lett. 45, 712–716 (1980)

11. Albano, A.M., Muench, J., Schwartz, C., Mees, A.I., Rapp, P.E.: Singular-value decomposition and the Grassberger-Procaccia algorithm. Phys. Rev. A 38, 3017–3026 (1988)
12. Fraser, A.M., Swinney, H.I.: Independent coordinates for strange attractors from mutual information. Phys. Rev. A. 33, 1134–1140 (1986)
13. Albano, A.M., Passamante, A., Farrell, M.E.: Using the higher-order correlations to define an embedding window. Physica D 54, 85–97 (1991)
14. Rosenstein, M.T., Collins, J.J., De Lucca, C.J.: Reconstruction expansion as a geometry-based framework for choosing proper delay times. Physica D 73, 82 (1994)
15. Kennel, M.B., Brown, R., Abarbanel, H.D.: Determining embedding dimension for phase-space reconstruction using a geometrical construction. Phys. Rev. A 45, 3403–3411 (1992)
16. Pope, J.E., Werner, S.S., Birkford, R.G.: Atlas of Neonatal Electroencephalography. Raven Press, New York (1992)
17. Janjarasjitt, S., Ocak, H., Loparo, K.A.: Bearing condition diagnosis and prognosis using applied nonlinear dynamical analysis of machine vibration signal. J. Sound and Vibration 317, 112–126 (2008)

# Solving Fuzzy Linear Regression with Hybrid Optimization*

M.H. Mashinchi[1], M.A. Orgun[1], and M. Mashinchi[2]

[1] Department of Computing, Macquarie University, Sydney, NSW, 2109, Australia
[2] Department of Statistics, Faculty of Mathematics and Computer Science, Shahid Bahonar University of Kerman, Iran

**Abstract.** Fuzzy linear regression is an important tool to find the linear inexact relationship between uncertain data. We then propose a hybrid optimization method based on tabu search and harmony search as a potential way of solving fuzzy linear regression. The proposed method aims at finding a model without considering any mathematical constraints while reducing the error of the regression's model in comparison to other methods. The experimental comparison of the results for two classes of crisp input-fuzzy output and fuzzy input-fuzzy output data sets shows the superiority of the method over the existing ones.

**Keywords:** Fuzzy linear regression, Tabu search, Harmony search, Hybrid optimization.

## 1 Introduction

Fuzzy Linear Regression Analysis (FLRA) is a tool to find the relationship between independant and depandant variables which comes with uncertainty. FLRA is a generalized form of classical regression in which all data can be represented by numerical values. The application of FLRA in environments that may be expressed with linguistic expressions has great potential [19,23].

Although Fuzzy Linear Regression (FLR) problems can be solved by a crisp linear regression analysis, a defuzzification process is needed to wash off the uncertainty implicit in the original data. The defuzzification phase makes the model to ignore some important information in fuzzy data [17].

There are two approaches for FLRA; possibilistic and Least Square (LS) [20]. In the former approach proposed by Tanaka *et al.* [22], the aim is to minimize the whole fuzziness by minimizing the total spreads of the fuzzy coefficients while the estimated outputs and the observed ones are within a certain $h$-level of belief [4]. The LS-based approach proposed by Dimond [6] aims at minimizing the least square error between the estimated and the observed fuzzy data.

In this paper, we propose a hybrid optimization method based on tabu search and harmony search for minimizing the error of the estimated regression model.

* This work has been supported in part under Australian Research Councils Linkage Projects Funding Scheme (project number LP0561985) and Macquarie University's International Travel Grant Scheme.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 336–343, 2009.

The ultimate goal of the porposed method is to find a fuzzy linear model which can fit the data with a smaller error compared to LS and probablistic approaches while no mathematical constraints are considered.

The rest of the paper is organized as follows. In Section 2, the mathematical formulation of FLRA is defined and then different approachs to solve a FLR are introduced. The weaknesses and strengths of each approach are mentioned as a motivation for proposing a hybrid optimization method based on tabu search and harmony search. In Section 3, the proposed method is discussed; we also show how to model a FLR as an optimization problem. The experimental results are given in Section 4. Finally Section 5 discusses the conclusions and future works.

## 2 Fuzzy Linear Regression

FLRA is a generalized form of the crisp regression model. In the crisp regression model, data come with no uncertainties and the error between the estimated and the observed outputs follows a normal distribution [4]. On the other hand, in some environments human subjective judgement may lead to imprecision which is due to fuzziness and cannot be justified with randomness [24].

In the FLR model, we are interested in finding a fuzzy function $\widetilde{y}$ given in (1) which can fit finite $k \epsilon \mathbb{N}$, crisp input-fuzzy output data $((x_{1i}, x_{2i}, \cdots, x_{(n-1)i}, x_{ni}), \widetilde{y}_i^*)$ (for $i = 1, \cdots, k$) with the minimum error [22].

$$\widetilde{y}_i = \widetilde{A}_n x_{ni} + \widetilde{A}_{n-1} x_{(n-1)i} + \cdots + \widetilde{A}_j x_{ji} + \cdots + \widetilde{A}_1 x_{1i} + \widetilde{A}_0 \qquad (1)$$

In a more general case, we aim to find a function given in (2) to fit the fuzzy input-fuzzy output data $\left((\widetilde{x}_{1i}, \widetilde{x}_{2i}, \cdots, \widetilde{x}_{(n-1)i}, \widetilde{x}_{ni}), \widetilde{y}_i^*\right)$ [18].

$$\widetilde{y}_i = \widetilde{A}_n \widetilde{x}_{ni} + \widetilde{A}_{n-1} \widetilde{x}_{(n-1)i} + \cdots + \widetilde{A}_j \widetilde{x}_{ji} + \cdots + \widetilde{A}_1 \widetilde{x}_{1i} + \widetilde{A}_0 \qquad (2)$$

where $\widetilde{A}_j$ (for $j = 0, \cdots, n$) are the fuzzy coefficients which should be computed based on the given input-output data such that the deviations between the observed outputs $\widetilde{y}_i^*$ and estimated ones $\widetilde{y}_i$ are the minimum possible value.

There are different ways of finding relationships within imprecise data. We can defuzzify all data into crisp data and then find a crisp relationship by crisp regression analysis. This is the simplest way of dealing with imprecision of data, though it neglects some useful information which is implicit in data. There are mainly two approaches for FLRA; probablistic and LS-based approaches [20]. The former finds a fuzzy relationship within a certain $h$-level of belief [4]. The term $h$ is the measurement fitness between the estimated fuzzy outputs and the observed ones [4]. The final error from the estimated error in these approaches has larger error compared to LS based approaches. However, the computational complexity of LS-based approaches has been reported high [13].

Meta-heuristic approaches have been also applied to find the best model which can fit the given data. To our knowledge, genetic algorithms [2,14], genetic programming [3] and artificial fuzzy neural networks [15] are among the meta-heuristics that have a good performance. The application of genetic algorithms and genetic programing is an efficient way, in case time is not a factor, as it is reported that these algorithms are slow to find a very accurate solution. In fact, these algorithms are very fast to find a near optimum solution but it takes a very long time to find a solution which is very close to the actual optimum solution [5]. Fuzzy neural networks are also good if the mathematical model of the regression is not sought as they do not show any model after the training process.

Considering the strenghs and weaknesses of all the approaches discussed above, we propose a hybrid optimization method based on tabu search and harmony search for FLRA. The main reasons for the application of the proposed hybrid optimization in FLRA are as follows;

- to decrease the estimated model's error compared to LS approaches;
- to have a constraint free model for solving FLRA. Constraints are a problem that arises in probablistic approaches which apply linear programming. Basically if more data is observed then additional constraints should be considerd which makes the linear programming sophisticated [1];
- to increase the speed of finding a very close optimum model compared to genetic algorithms and genetic programing; and
- returning the actual mathematical formula of the estimated model, which is not obtainable with fuzzy neural networs.

## 3 Hybrid Optimization for FLRA

Meta-heuristics, in comparison with mathematical approaches, are famous for being context-independent. This is because they deal with the problem as a black box and there are no mathematical constraints in their application. As discussed above, some meta-heuristics such as tabu search and genetic algorithms are fast to find a near optimum solution but slow to seek a solution close to the optimum solution [5]. To increase the speed and the goodness of the solution they are hybrized with local optimizer methods (LOMs) [5].

Meta-heuristics can be hybridized with LOMs in different ways; for a comprehensive study one can refer to [21]. A common way of hybridizing is to start diversification of the search space followed by an inensification process. As the diversification phase is a very expensive task, it can be stopped quicker. However, a very early switch to intensification increases the possibility of not finding the global optimum. Conversely, a very late switch to intensification makes diversification longer and consequently makes the optimization process very time consuming [16].

There are two major ways of hybridization. One way is to run diversification for some time and then intensification proceeds with the best solution found in the diversification phase. This approach is very fast but the possibility of finding a local minimum is high. The other approach is to do intensification for each of the potential solutions which are found by the diversification phase. This hybrid approach is very time consuming as the intensification phase is carried out for each potential solution. However, this is a safer approach to find a global optimum solution. In this paper, we combine these two approaches. First, the diversification phase is done by tabu search. In this phase all potential solutions are stored and then an intermidiate phase, known as semi-intensification, explores which one is the best one. The semi-intensification has a more close look on the potential solutions' surrounding area. Finally the best solution is intensified comprehensively by harmony search.

Tabu search is applied in the diversification phase as it has more focus on escaping the local minima. It starts from a random solution and copies it in the current-solution ($C$) and the best-solution ($S^*$). Then some neighbours are generated randomly based on a neighbour-search strategy around $C$ and ranked according to their performance. Then the best neighbour is selected and copied to $C$ if it has not been tabu based on the tabu list. A tabu neighbor can be still selected as the next move, having the condition of better performace compared to $S^*$. This condition is known as *the aspiration condition.* Tabu search differs from hill climbing methods in a way that it can move from a better solution to a worse solution [8].

Harmony search is a meta-heuristic approach which adopts the idea of natural musical processes [12]. The goal of harmony search is to minimize an objective function like $f(x)$ where $f : \mathbb{R}^n \rightarrow \mathbb{R}$. In harmony search, some randomly generated feasible solutions are initialized in the harmony memory. In each iteration, the algorithm aims at improvising the harmony memory. The improvision process works based on three operations; memory consideration, pitch adjustment and random selection. In the memory consideration and the random selection, each variable of a new solution vector is generated either taken from the harmony memory or selected randomly. The pitch adjustment operation makes random changes to some of the generated solutions. This operation has been introduced to escape from local minima [12].

Tabu search is a point to point meta-heuristic approach [8]. In contrast to population based methods (e.g. genetic algorithms and swarm optimization), it does not need a large number of solutions in each iteration. The total number of function evaluations is reduced, since not many solutions are required for searching the neighborhood of the current solution. However, to be able to search all around the search space, the solutions met already are put into the tabu list for not searching them again later. Tabu search has good performance in diversification due to its tabu list, but, if a near optimal solution is found, then we should switch to the intensification phase. The aim of hybridizing tabu search with an LOM is to decrease the total number of function evaluations and consequently increasing the speed of the optimization process.

The next step is to model the FLRA as an optimization problem which can be solved by the proposed method. The error function for an estimated model over the given data, can be defined as $e_i = f(\widetilde{A}_i)$. So different $\widetilde{A}_i$ gives different $e_i$, and we are looking for $\widetilde{A}_i$ where total $e_i$ for all given data ($i = 1, \cdots, k$) is the minimum possible value. Without loss of generality for a fuzzy value like $\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x))|\ x\epsilon\mathbb{R},\ \mu_{\widetilde{A}} \rightarrow [0, 1]\}$, we can represent it as $\widetilde{A} = (a_i^l, a_i^c, a_i^r)$ if it is a triangular fuzzy value. The parameters $a_l$, $a_r$ and $a_c$ represent the left and the right spreads and the center of the fuzzy value $\widetilde{A}$, respectively. Thus the minimization problem which should be solved is as (3).

$$\min \sum_{i=1}^{n} f_i(a_n^l, a_n^c, a_n^r, \cdots, a_0^l, a_0^c, a_0^r) \tag{3}$$
$$f_i : \mathbb{R}^{3n} \to \mathbb{R}$$

One of the common error objective functions in FLRA is the none-intersected area between the observed and the estimated output [1,7,9,10,11,20].

This error value for $i^{th}$ given input-output is given in (4). In (4), $\widetilde{y}^*$ and $\widetilde{y}$ are the observed and estimated fuzzy outputs, respectively.

$$f_i = \left[ \int_{-\infty}^{+\infty} |\widetilde{y}_i(x) - \widetilde{y}_i^*(x)| dx \right] \tag{4}$$

## 4 Simulations and Results

In this section, we discuss our experiments with the hybrid optimization method on two well-known data sets. For both data sets, search is carried out for 300 iterations for all of diversification, semi-intensification and intesification. In each iteration only 10 neighbors are considered. The stopping criterion is set either reaching the maximum iterations or not having any improvement for 10 consecutive iterations in each phase. For the sake of fairness, 100 separate runs are cosidered for each of the data sets. Each run starts with different random solutions and seeds. Then the best error, average error and standard deviation are reported for each data set.

### 4.1 Crisp Input-Fuzzy Output Example

The first data set is an example of crisp input-fuzzy output data and was introduced by Tanaka *et al.* in the first FLRA article [22], see Table 1. This data set has been applied as a benchmark data set in many works reported in the literature [7,9,10,20]. After 100 separate runs, the best estimated model computed by the proposed method is $\widetilde{y} = (5.07, 6.72, 8.38) + (1.12, 1.27, 1.38)x$. The error of this model based on the error measurement given in (4) is 8.51. The average error of 100 runs is 8.87 with the standard deviation of 0.26. According to comparison results given in Table 2 our method has the smallest error. The best estimated function is depicted in Figure 1.

### 4.2 Fuzzy Input-Fuzzy Output Example

The second data set is an example of a more generalized case where both inputs and outputs are fuzzy values [18], Table 3. The best estimated model by

**Table 1.** Tanaka's dataset [22]

| Obs. | Independent variable | Dependent variable |
|---|---|---|
| 1 | 1 | $(6.2, 8.0, 9.8)$ |
| 2 | 2 | $(4.2, 6.4, 8.6)$ |
| 3 | 3 | $(6.9, 9.5, 12.1)$ |
| 4 | 4 | $(10.9, 13.5, 16.1)$ |
| 5 | 5 | $(10.6, 13.0, 15.4)$ |

**Table 2.** Comparison of estimated model's error for the data given in Table 1

| | This work | Hassanpour [7] | Tanaka [22] | Kao [9] | Shakouri [20] |
|---|---|---|---|---|---|
| Total error | 8.51 | 9.165 | 12.4017 | 9.6776 | 12.9693 |

**Table 3.** Sakawa's dataset [18]

| Obs. | Independent variable | Dependent variable |
|---|---|---|
| 1 | $(1.5, 2, 2.5)$ | $(3.5, 4.0, 4.5)$ |
| 2 | $(3.0, 3.5, 4.0)$ | $(5.0, 5.5, 6.0)$ |
| 3 | $(4.5, 5.5, 6.5)$ | $(6.5, 7.5, 8.5)$ |
| 4 | $(6.5, 7.0, 7.5)$ | $(6.0, 6.5, 7.0)$ |
| 5 | $(8.0, 8.5, 9.0)$ | $(8.0, 8.5, 9.0)$ |
| 6 | $(9.5, 10.5, 11.5)$ | $(7.0, 8.0, 9.0)$ |
| 7 | $(10.5, 11.0, 11.5)$ | $(10.0, 10.5, 11.0)$ |
| 8 | $(12.0, 12.5, 13.0)$ | $(9.0, 9.5, 10.0)$ |

**Table 4.** Comparison of estimated model's error for the data given in Table 3

| | This work | Sakawa92 [18] | Kao [10] | Kao [9] | Hassanpour [7] | Arabpour [1] |
|---|---|---|---|---|---|---|
| Total error | 5.33 | 9.43 | 9.36 | 7.47 | 5.74 | 7.69 |

our method for 100 separate runs is $\widetilde{y} = (3.00, 3.02, 3.20) + (0.5, 0.5, 0.5)\widetilde{x}$. The comparison of the estimated errors for different methods is given in Table 4. The average error and standard deviation for 100 separate runs are 5.64 and 0.38, respectively. According to the result given in Table 4, the proposed hybrid optimization method outperforms the other methods.

## 5 Conclusions and Future Works

This paper proposed a hybrid optimization method based on tabu search and harmony search for fuzzy linear regression analysis. The diversification of this

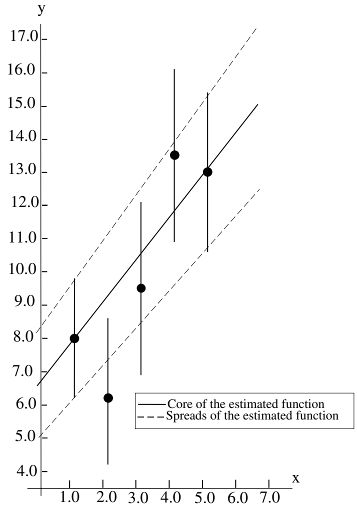


**Fig. 1.** The estimated fuzzy function for the given data in Table 1

method is carried out by tabu search and then harmony search does the intensification. There are two main advantages compared to probabilistic and least square approaches. First, we do not need to consider any mathematical constraints, a condition that exists in probabilistic approaches which apply linear programming. In these approaches the number of constraints increases provided that more data is considered. Also, the model estimated based on the proposed method has smaller error compared to both probablistic and LS-based approaches. Based on the benchmark data sets, our method outperforms the other methods reported in the literature.

We plan to apply the hybrid method to a class of none-parametric fuzzy regression problems. There are few works dedicated to this class of fuzzy regression. We also plan to consider the application of tabu programming as an optimization method which is capable of finding solutions with different structures.

## References

1. Arabpour, A.R., Tata, M.: Estimating the parameters of a fuzzy linear regression model. Iranian Journal of Fuzzy Systems 5(2), 1–19 (2008)
2. Buckley, J.J., Feuring, T.: Linear and non-linear fuzzy regression: evolutionary algorithm solutions. Fuzzy Sets and Systems 112(3), 381–394 (2000)
3. Chan, K.Y., Kwong, C.K., Tsim, Y.C.: A genetic programming based fuzzy regression approach to modelling manufacturing processes. International Journal of Production Research, 1–16 (2009)
4. Chang, Y.-H.O., Ayyub, B.M.: Fuzzy regression methods - a comparative assessment. Fuzzy Sets and Systems 119(2), 187–203 (2001)

5. Chelouah, R., Siarry, P.: Genetic and nelder-mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. European Journal of Operational Research 148(2), 335–348 (2003)
6. Diamond, P.: Least squares fitting of several fuzzy variables. In: Bezdek, J.C. (ed.) Analysis of Fuzzy Information, pp. 329–331. CRC Press, Tokyo (1987)
7. Hassanpour, H.: A Goal Programming Approach to Fuzzy Linear Regression. PhD thesis, Shahid Bahonar University of Kerman (2008)
8. Hedar, A.-R.: Studies on Metaheuristics for Continuous Global Optimization Problems. PhD thesis, Kyoto University (2004)
9. Kao, C., Chyu, C.-L.: A fuzzy linear regression model with better explanatory power. Fuzzy Sets and Systems 126(3), 401–409 (2002)
10. Kao, C., Chyu, C.-L.: Least-squares estimates in fuzzy regression analysis. European Journal of Operational Research 148(2), 426–435 (2003)
11. Kim, B., Bishu, R.R.: Evaluation of fuzzy linear regression models by comparing membership functions. Fuzzy Sets and Systems 100(1-3), 343–352 (1998)
12. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Computer Methods in Applied Mechanics and Engineering 194(36–38), 3902–3933 (2005)
13. Lu, J., Wang, R.: An enhanced fuzzy linear regression model with more flexible spreads. Fuzzy Sets and Systems (to appear)
14. Mashinchi, M.H., Mashinchi, M.R., Mashinchi, M.: A two-phased genetic algorithm approach for linear and quadratic fuzzy regression. Set-valued Mathematics and Applications 1(2), 183–190 (2008)
15. Mashinchi, M.R., Selamat, A.: An improvement on genetic-based learning method for fuzzy artificial neural networks. Applied Soft Computing 9(4), 1208–1216 (2009)
16. Meittinen, K., Makela, M.M., Maaranen, H.: Efficient hybrid methods for global continuous optimization based on simulated annealing. Computers and Operations Research 33(4), 1102–1116 (2006)
17. Näther, W.: Regression with fuzzy random data. Computational Statistics & Data Analysis 51(1), 235–252 (2006)
18. Sakawa, M., Yano, H.: Fuzzy linear regression and its applications. Fuzzy Regression Analysis, pp. 61–80. Physica-Verlag, Heidelberg (1992)
19. Sánchez, J.A., Gómez, A.T.: Applications of fuzzy regression in actuarial analysis. Journal of Risk Insurance 70, 665–669 (2003)
20. Shakouri, H., Nadimia, R.: A novel fuzzy linear regression model based on a non-equality possibility index and optimum uncertainty. Applied Soft Computing 9(2), 590–598 (2009)
21. Talbi, E.-G.: A taxonomy of hybrid metaheuristics. Journal of Heuristics 8(5), 541–564 (2002)
22. Tanaka, H., Uejima, S., Asal, A.: Linear regression analysis with fuzzy model. IEEE Transactions on Systems, Man and Cybernetics 12, 903–907 (1982)
23. Xue, Y., Kim, I.S., Son, J.S., Park, C.E., Kim, H.H., Sung, B.S., Kim, I.J., Kim, H.J., Kang, B.Y.: Fuzzy regression method for prediction and control the bead width in the robotic arc-welding process. Journal of Materials Processing Technology 164-165, 1134–1139 (2005)
24. Yang, M.-S., Ko, C.-H.: On cluster-wise fuzzy regression analysis. IEEE Transactions on Systems, Man, and Cybernetics, Part B 27(1), 1–13 (1997)

# Automatic Document Tagging in Social Semantic Digital Library

Xiaomei Xu and Zhendong Niu

School of Computer Science,
Beijing Institute of Technology, Beijing, 100081, PRC
{gloria_xu,zniu}@bit.edu.cn

**Abstract.** The emergence of *Web 2.0* has created a lot of annotation and personalization information about web resources. Extracting and utilizing these information to enhance the quality of services is a key target of modern digital libraries. In this paper, we present a novel Automatic Document Tagging (*ADT*) approach for digital libraries. In our approach, the *ADT* problem is formulated as a variant of multi-class classification problem. But differently, the training data for *ADT* is collected from the user's historic tags and only partially labeled. The incompleteness of the training data makes the training a more challenging problem. To overcome this problem, an efficient randomized online training algorithm (*RPL*) is proposed. *RPL* algorithm has two phases: (i) random exploitation and (ii) classifier update. The experimental results from both synthetic and real-word data demonstrate the effectiveness.

**Keywords:** Digital library, information retrieval, semantic web, ontology, multi-class classification, online learning.

## 1 Introduction

In the past decades, digital library went through a rapid development as the fast growth of computer technologies and IT industry. Nowadays, it is one of the most important means for E-learning and self-education. There are a lot of research efforts focusing on how to design the digital library system [5][8] and many practical large digital libraries come out of those researches, such as the Alexandria digital library from University of California at Santa Barbara, Greenstone digital library and Google Book Search Project.

Recently, *Web 2.0*, Semantic Web and Social Networks emerged and became more and more popular [7][15]. Those technologies create many innovative ways to use the web and revolutionize the designs of web applications, i.e. metadata management, semantic search and browsing, personalized and community-aware services. Kruk et.al proposed the concept of social semantic digital library in [10][11]. In general, the semantic digital libraries deliver services that combine semantic web and social networking technology with the digital library system.

In this paper, we focus on the particular problem of Automatic Document Tagging (*ADT*) in semantic digital library. A typical usage for *ADT* is as follows. When a user browses a digital library document that appears to be valuable

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 344–351, 2009.

to him/her, he/she would bookmark and tag the document by his/her interest. With this information, the $ADT$ will learn the interest for this user and intelligently reorganize the tags for the library documents and search the relevant documents according to the user's interest in future.

## 2 Automatic Document Tagging (*ADT*)

In this section, we present an overview the problem description of automatic document tagging ($ADT$) for semantic digital library in the setting of online multi-class classification.

$ADT$ can be described as follows: given a set of documents $\mathbf{D} = \{\mathbf{d_1}, \mathbf{d_2}, \cdots, \mathbf{d_n}\}$ and a set of tags $T = \{t_1, t_2, \cdots t_m\}$, the task of $ADT$ is to find a mapping between the document and its appropriate tags. In this setting, the $ADT$ problem can be regarded as a variant of online multi-class classification problem [1][2][4].

However, there is a significant difference in the characteristics of the training data between these two problems. In traditional online multi-class classification, the training data are complement. For every training sample, the true class label (possibly more than one or none at all) is given. But for our $ADT$ in semantic digital library, it's incompletely labeled training data since the training data is collected from the historic tags in the users' previous browsing sessions and annotations.

The online learning for $ADT$ is performed in a sequence of consecutive prediction/training rounds from the user's browsing sessions. During the $i_{th}$ round, the $ADT$ system is presented with a document instance in its feature vector representation $x_i \in \mathbb{R}^d$ and predicts the label from a set of $k$ predefined labels. The predefined set of labels are denoted as $[\mathbf{k}] = \{1, \cdots, k\}$ and the predicted label is denoted as $\hat{y}_i \in [\mathbf{k}]$. In traditional multi-class classification scenario, after predicting the label, the classifier directly receives the feedback of the true label for $x_i$, denoted by $y_i \in [\mathbf{k}]$. But in our $ADT$ scenario, the feedback received is different. The $ADT$ receives a "yes or no" to whether the prediction is correct instead of the true label. Mathematically, the feedback is represented as:

$$f(y_i; \hat{y}_i) = \mathbf{1}[\hat{y}_i = y_i] \tag{1}$$

where $\mathbf{1}[p]$ is equal to 1 if predicate $p$ holds and 0 otherwise. The objective of the learning algorithm is to minimize cost function denoted by equation (2) when running through the training/prediction sequence:

$$e = \sum_{i=1}^{N} \mathbf{1}[\hat{y}_i \neq y_i] \tag{2}$$

where $e$ is the sum of number of prediction errors. To minimize $e$, $ADT$ greedily updates its prediction mechanism after each round in hope that it will make fewer mistakes for future rounds. Let's say the prediction algorithm at round $i$ is determined by a hypothesis classifier:

$$\mathbf{h}_i : \mathbb{R}^d \Rightarrow [\mathbf{k}] \tag{3}$$

where $\mathbf{h}_i$ is taken from a set of hypotheses classifiers $\{\mathcal{H}\}$. Fast query response is very important for *ADT* in social semantic digital library. Therefore, we focus on the set of linear hypotheses classifiers in this paper as the simplicity of linear hypothesis classifier leads to efficient learning algorithm with large data. Specifically, each $\mathbf{h}_i \in \mathcal{H}$ is parameterized by a weight matrix $\mathbf{W} \in \mathbb{R}^{k \times d}$. The classifier hypothesis is defined to be

$$\mathbf{h}(x) = \underset{j \in [\mathbf{k}]}{\operatorname{argmax}} (\mathbf{W} \cdot x)_j \tag{4}$$

where $(\mathbf{W} \cdot x)_j$ is the $j_{th}$ element of the predicted score vector. Each row in $\mathbf{W}$ represents a weight vector for a particular tag. The predicted tag is chosen to be the one with the highest score. The performance of *ADT* is determined by the weight matrix $\mathbf{W}$. If $y \neq h(x)$ for a given sample $(x; y)$, the *ADT* makes a prediction mistake. The core problem is to find an efficient algorithm to learn the weight matrix ($\mathbf{W}$).

# 3 Online Learning Algorithm

## 3.1 Perceptron Learning Algorithm

Perceptron algorithm [3][18] is a simple learning algorithm. It uses the simplest linear classifier and is capable of classifying the linearly separatable data. Despite its simplicity, *PL* algorithm [17] has been proved to be quite effective for solving large practical problems [6]. The learning algorithm is modeled as the process of updating the weight matrix across the iterations. The weight matrix is denoted by $\mathbf{W}_i$ for the $i_{th}$ iterations. For every input training sample, we update $\mathbf{W}$ as follows:

$$\mathbf{W}_{i+1} = \mathbf{W}_i + \alpha U_i \cdot x_i^T \tag{5}$$

where $\alpha$ is the step size for the update, $x_i$ is the training sample input vector, $U_i \in R^{k \times 1}$ is the update vector. The update vector is defined as

$$U_i = [I(y_i) - I(\hat{y}_i)] \tag{6}$$

where $I(k)$ is an indicator vector with the $k_{th}$ element equal to 1. If the prediction is correct (i.e. $y_i = \hat{y}_i$), then there is no update (i.e. $U = 0$ and $\mathbf{W}_{i+1} = \mathbf{W}_i$). If there is a misprediction, then $x_i$ is added to the $y_i$th row of $\mathbf{W}$ to boost the weight for the correct tag and subtracted from the $\hat{y}_i$th row of $\mathbf{W}$ to suppress the incorrect tag. Despite of its simplicity, the *PL* algorithm has been widely used for many applications, especially for learning with large data.

## 3.2 Randomized Perceptron Learning Algorithm

*PL* algorithm learns from completely labeled training data. Randomized Perceptron Learning (*RPL*) Algorithm learns from the incomplete data for the *ADT* problem. Similar to *PL*algorithm, we use the current weight matrix to predict the best label for the sample:

$$\hat{y}_i = \underset{j \in [\mathbf{k}]}{\text{argmax}} (\mathbf{W} \cdot x)_j \tag{7}$$

where **W** is the weight matrix, and $x$ is test sample.

If the prediction is correct, the weight matrix (**W**) gets updated in the same way as the *PL* algorithm does. But if the prediction is incorrect (i.e. $\hat{y}_i \neq y_i$) we can't make the same update as the *PL* algorithm as we don't know the true tag label for the sample. For this reason, we introduce the random prediction phase inside our *RPL* algorithm. On the round when the misprediction happens, the *PRL* algorithm continues exploring other tag categories with a certain probability defined as follows:

$$P(r) = (1 - \gamma)\mathbf{1}[r = \hat{y}_i] + \frac{\gamma}{size(s) - 1} \tag{8}$$

where the parameter ($\gamma$) controls the exploration-exploitation trade-off. The larger the $\gamma$, the higher the probability we are in the exploration stage and vice versa. In this way, we indirectly obtain the information regarding the true tag label ($y_i$) for the sample through randomized exploration. The weight matrix (**W**) is updated with binary feedbacks from the exploration. The *RPL* algorithm is summarized in Algorithm 1. The repeat-loop nested inside the for-loop performs the random exploration. The weight matrix **W** gets updated in line 8 and line 12 during the exploration.

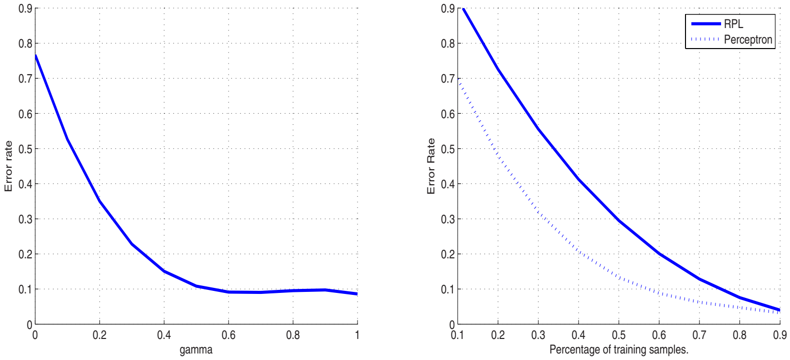
## 4 Experimental Result

In this section, we present the experimental results for Randomized Perceptron Learning (*RPL*) Algorithm on both synthetic and real world data. The first data set, denoted by *SS1*, is a synthetic nine-class, 300-dimensional data of size 1000. The coordinates in the vectors represent different words of a small vocabulary of size 300. The 9 class synthetic data sets are generated as follows. First, we choose 9 50-dimension vectors as centroids: $v_1, \cdots, v_9$. The centroid vector $v_i$ corresponds to the $i_{th}$ topic in the document database. The 50 coordinates correspond to the keywords in the data set. To generate a sample documents, we select one topic vector ($v_i$) in random and add gaussian ($s_i = v_i + n$). We use $s_i$ for the first 50 coordinates of the sample vector ($d(1 : 50) = s_i$). Uniform random variables are used to fill in the remaining 250 coordinates. In this way, the remaining 250 coordinates model the common words, which can appear in any document from any topic.

Figure 1(a) reports the average error rate for *RPL* with different $\gamma$ over 10 runs. The x-axis shows the $\gamma$ value and the y-axis shows the error rate. As $\gamma$ goes up, the error rate drops. But after $\gamma$ goes above a certain value ($\gamma \geq 0.5$), the error rate is stabilized. This indicates that we have reached the maximal benefit of random exploration in *RPL*. Figure 1(b) compares the error rate between *RPL*

**Algorithm 1.** Randomized Perceptron Learning (RPL) Algorithm

```
Parameters: $\alpha \in [0.1, 0.9], \gamma \in (0, 1)$
Initialize weight matrix, $\mathbf{W}_0 = 0 \in \mathbb{R}^{k \times d}$.
for $i = 1$ to $N$ do
    Predict the class label for the $i_{th}$ sample , $\hat{y}_i = \underset{j \in [\mathbf{k}]}{\text{argmax}}(\mathbf{W} \cdot x)_j$.
    define the candidate label set $s = [\mathbf{k}]$
    repeat
        Calculate the sample probability: $P(r) = (1 - \gamma)\mathbf{1}[r = \hat{y}_i] + \frac{\gamma}{size(s) - 1}$.
        if Receive positive feedback, i.e. "clicked" then
            Update $\mathbf{W}$: $\mathbf{W}_{i+1} = \mathbf{W}_i + \alpha[1 - P(\hat{y}_i)]I(\hat{y}_i) \cdot x_i^T$
            Stop exploration.
        else
            Randomly choose one label $\widetilde{y}$ according to probability $P(r)$ from the label set $s$.
            Update $\mathbf{W}$: $\mathbf{W}_{i+1} = \mathbf{W}_i - \alpha P(\widetilde{y}_i)I(\widetilde{y}_i) \cdot x_i^T$
            Update $\alpha$: $\alpha = P(\widetilde{y}) \cdot \alpha$.
            Remove the mispredicted $\hat{y}$ from the candidate set, $s = s - \{\hat{y}_i\}$
        end if
    until Exceed the exploration limit
end for
```



(a) (b)

**Fig. 1.** Comparison of *RPL* v.s. *PL* on *SS1* data. (a) Error rate for *RPL* with different exploration parameter $\gamma$. (b) Comparison of error rate between *RPL* and *PL* algorithms.

($\gamma = 0.5$) and the ideal $PL$. At the beginning of training, the $RPL$ performs worse than the $PL$. The error rate of $RPL$ is 20% higher than the $PL$ algorithm for small percentage of training data. As the training continues, the error rate from $RPL$ drops at slightly faster rate than the $PL$ algorithm. Towards the completion of training, the $RPL$ algorithm performs very close to the $PL$ algorithm. This demonstrates that the $RPL$ can achieve the performance close to the $PL$ in the long run.

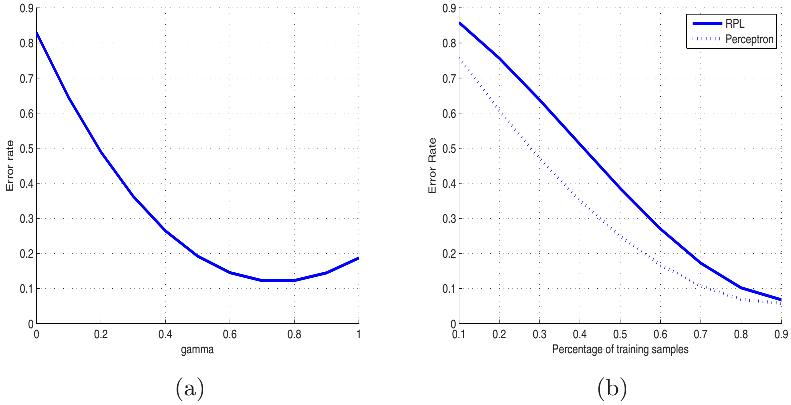
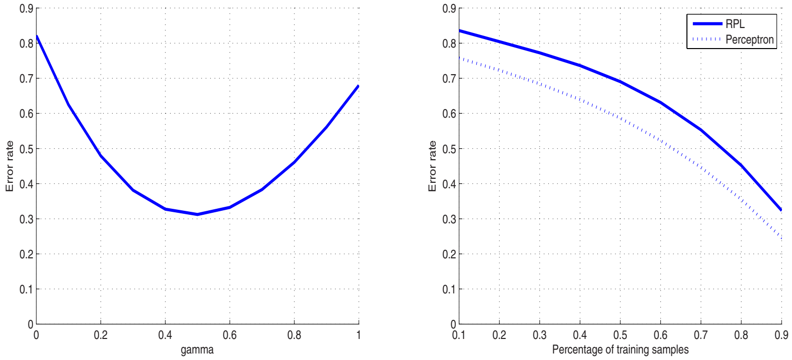


**Fig. 2.** Comparison of $RPL$ v.s. $PL$ on $SS2$ data. (a) Error rate for $RPL$ with different exploration parameter $\gamma$. (b) Comparison of error rate between $RPL$ and $PL$ algorithms.

The second data set, denoted by $SS2$, is constructed in the same way as $SS1$ except for the additional 5% label noise. The label noise is used to model the uncertainty and mistakes made by the user when tagging the document. This makes the learning more challenging. Figure 2(a) shows the average error rate from $RPL$ with different $\gamma$ over 10 runs. The result is similar to what is shown in Figure 1(a). It is interesting to notice that larger $\gamma$ actually increase the error rate and perform worse. This is because label noise may pollute the learner and make it less accurate. It indicates that setting $\gamma$ too low or too high leads to higher error rates when facing with label noise. For specific application it may be desirable to have a dynamic adjustable $\gamma$ in the process of training. Figure 2(b) compares the error rate for $RPL(\gamma = 0.65)$ with the $PL$. The error rate of $RPL$ is 10% higher than Perceptron algorithm initially. With the progress of the training, the error rate drops for both $RPL$ and $PL$. The margin between them stay constant. After the 50% of the training data, the error rate of $RPL$ drops faster. Towards the completion of training, $RPL$ reaches almost the same error rate as the $PL$ algorithm.

The third data set comes from real-world text documents. The data set is prepared in a similar way as the real-world text classification task [12]. The documents are collected from different biomedical engineering journals. Documents



(a) (b)

**Fig. 3.** Comparison of *RPL* v.s. *PL* on *SS3* data. (a) Error rate for RPL with different exploration parameter $\gamma$. (b) Comparison of error rate between RPL and Perceptron algorithms.

in the data set are tagged with 7 different categories, i.e. genetics, oncology psychiatry, dental, heart disease, brain disease, medical imaging. Figure 3(a) shows the error rate for *RPL* against the exploration parameter $\gamma$. Figure 3(b) compares the error rate for $RPL(\gamma = 0.5)$ with the *PL*. Compared with the previous two synthetic data, the error rate for both *RPL* and *PL* algorithm are higher. The final error rates for *PL* and *RPL* are 26% and 32% respectively. However, it is clear from the plot that as the number of examples grows, the error rate of *RPL* is dropping at a rate comparable to that of *PL*. This demonstrate that the performance of *RPL* with incomplete data is comparable for that of traditional *PL* with complete data.

## 5 Conclusion and Future Works

In fact, there are many related works on solving the problem of online learning with partial label information [4][9][13]. [9] proposes a learning algorithm using online convex optimization , but it doesn't apply to the problem of using non-convex classification loss function and requires good gradient estimation. [13] improves *PL* algorithm for the online learning algorithm in a more general setting where the learner has side information. However, its too general settings lead to less efficiency. Besides, its i.i.d (identity independent distribution) assumption of the problem doesn't hold for the *ADT* problem.

Our algorithm is derived directly from the simple *PL* algorithm and makes no statistical assumption on the data. We use a linear hypothesis to model the problem and a straightforward multiplicative update scheme. This leads to the efficient and practical algorithm for large data in real-world *ADT* problem. Experimental results on both synthetic and real-world data show (*RPL*) achieves good performance. In future, we will investigate the extension of (*RPL*)

[19][13][9] with other web search technologies [16][14] and apply them to other open problems in semantic digital libraries.

## References

1. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive aggressive algorithms. Journal of Machine Learning Research 7, 551–585 (2006)
2. Crammer, K., Singer, Y.: Ultraconservative online algorithms for multiclass problems. Journal of Machine Learning Research 3(951) (2003)
3. Duda, R., Hart, P.: Pattern Classification and Scene Analysis. Wiley, Chichester (1973)
4. Fink, M., Shalev-Shwartz, S., Singer, Y., Ullman, S.: On- line multiclass learning by interclass hypothesis sharing. In: Proceedings of the 23rd International Conference on Machine Learning (2006)
5. Fox, E.: The digital libraries initiative - update and discussion. In: Bulletin of the America Society of Information Science, October/November 1999, vol. 26 (1999)
6. Freund, Y., Schapire, R.: Large margin classification using the perceptron algorithm. Machine Learning 37(3), 277–296 (1999)
7. Geroimenko, V.: A semantic web primer. Computer Journal 48(1) (2006)
8. Kahn, R., Cerf, V.: An open architecture for digital library system and a plan for its development. Digital Libary Project 1 (1998)
9. Kivinen, J., Warmuth, M.: Exponentiated gradient versus gradient descent for linear predictors. Information and Computation 132 (January 1997)
10. Kruk, S.R., Decker, S., Zieborak, L.: JeromeDL - adding semantic web technologies to digital libraries. In: Andersen, K.V., Debenham, J., Wagner, R. (eds.) DEXA 2005. LNCS, vol. 3588, pp. 716–725. Springer, Heidelberg (2005)
11. Kruk, S., Woroniecki, T., Gzella, A., Dabrowski, M., McDaniel, B.: Anatomy of a social semantic library. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519. Springer, Heidelberg (2007)
12. Lang, K.: Newsweeder: Learning to filter netnews. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 331–339. Morgan Kaufmann, San Francisco (1995)
13. Langford, J., Zhang, T.: The epoch-greedy algorithm for contextual multi-armed bandits. In: NIPS (2007)
14. Langville, A., Carl, D.: Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton University Press, Princeton (2006)
15. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 522–536. Springer, Heidelberg (2005)
16. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web (1999)
17. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 65, 386–407 (1988)
18. Vapnik, V.: Statistical Learning Theory. Wiley, Chichester (1998)
19. Weston, J., Watkins, C.: Support vector machines for multi-class pattern recognition. In: Proceedings of the Seventh European Symposium on Artificial Neural Networks (April 1999)

# Text Mining with an Augmented Version of the Bisecting K-Means Algorithm

Yutaro Hatagami and Toshihiko Matsuka

Department of Cognitive and Information Science
Chiba University, Chiba, Japan

**Abstract.** There is an ever increasing number of electronic documents available today and the task of organizing and categorizing this ever growing corpus of electronic documents has become too large to perform by analog means. In this paper, we have proposed an augmented version of the bisecting k-means clustering algorithm for automated text categorization tasks. In our augmented version, we have added (1) a bootstrap aggregating procedure, (2) a bisecting criteria that relies on dispersions of data within clusters, and (3) a method to automatically terminate the algorithm when an optimal number of clusters have been produced. We have performed text categorization experiments in order to compare our algorithm against the standard bisecting k-means and k-means algorithms. The results showed that our augmented version improved approximately 15% and 20% in classification accuracies compared to the standard bisecting k-means and k-means, respectively.

## 1 Introduction

Due to the rapid growth of the Internet and the advancements in computer technologies, there is an ever increasing number of electronic documents available to people all over the world. The task of organizing and categorizing this continually growing corpus of electronic documents has become much too large in size to perform by any analog means. As a result, the need for an automated means of processing these corpora and research on text mining, a type of data mining specialized for text data, have kept increasing over the past several years.

In the field of text mining, there are two ways to process documents into categories. One way is a supervised approach, and this approach is often referred to as classification. The task of classification involves associating properly labeled documents with categories in order to identify rules and patterns for document classification. When presented with new unlabeled data, the trained classifier can classify the data into the categories by applying the rules. The other approach is an unsupervised method, and it usually involves identifying groups of "coherent" documents in a data set. This process is often referred to as clustering. Clustering does not require properly labeled documents as its training data set. Rather, by using the similarity among documents, it identifies coherent groups among unlabeled documents.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 352–359, 2009.

In this paper we will focus on the later approach and propose what we call an augmented bisecting k-means clustering (ABKM) algorithm. We, then, perform experiments in order to evaluate the effectiveness of the augmented algorithm.

### 1.1 Bisecting K-Means Algorithm

One of the most popular clustering methods used today is the k-means clustering algorithm. The k-means clustering algorithm, known for its simplicity, is used in many fields for pattern recognition and cluster analysis. However, it has been reported that the bisecting k-means algorithm, an augmented variant of the original k-means algorithm, produces better clustering results than the standard k-means in text mining [2]. The bisecting k-means simply repeats (1) standard k-means clustering where $k$ is fixed at 2 and then (2) split one, typically the largest, cluster in to two clusters until the desired number of clusters is produced. Detailed descriptions and rationale of the bisecting k-means algorithm can be found in [2].

**Disadvantages of the Algorithm.** The bisecting k-means algorithm is known to produce better results than the original k-means algorithm, yet there is still room for improvement. For instance, as in the original k-means, the initial centroids are usually chosen randomly in a typical bisecting k-means clustering. This in turn may result in obtaining unstable and unreliable clusters and cluster centroids. In other words, initial starting points heavily influence the outcome of the analysis, and thus the results of analyses are not always identical nor optimal. Another known setback of the k-means algorithm is that the number of clusters to be made must be determined a priori. In real world problems, the optimal number of clusters is hard to know. It requires multiple runs in order to determine an optimal number of clusters (e.g. determining it by examining a scree plot). The third disadvantage with the bisecting k-means algorithm is that it implicitly assumes that clusters are similar in their sizes. That is, the algorithm usually selects the largest cluster or the cluster with the largest sum of squared error(SSE) value to be split. Thus a bigger cluster (i.e., cluster with the largest number of instances) is almost always chosen to be divided into two smaller clusters. This is not a problem for a data set that can be split into clusters that are similar in their sizes. However, with real world data, we cannot always expect the number of instances in every cluster to be similar.

## 2 Improving Bisecting K-Means

In this section we will present computational procedures aimed at improving the original bisecting k-means algorithm by resolving the problems presented in the previous section. The first procedure we considered was augmenting a bootstrap aggregating process to resolve the instability of the k-means algorithm caused by the random selection of initial centroids. Secondly, we modified the criterion for identifying a cluster to be bisected. Instead of using the cluster size or aggregated distances between centroids and data points (i.e., SSE), we used each cluster's standard deviation as a criterion for bisection. Lastly, in order to address the problem of determining the number of clusters a priori, we incorporated an automated process to terminate analysis when an "optimal" number of clusters have been created.

### 2.1 Bootstrapping Aggregating Method

In order to achieve stable and reliable clustering results, we included a bootstrap aggregating procedure in the k-mean clustering step of our bisecting k-means. In other words, we added a step to repeat the k-means clustering process for a certain number of times and then consider the median of the resulting centroids as the "true" centroids of the clusters. By aggregating results of repetitive k-means analyses, the effect of random initialization would be reduced and more optimal centroids should be identified.

### 2.2 Criterion for Bisection

Because the standard bisecting k-means almost always bisect the largest cluster into two smaller ones, the produced clusters are usually uniform in size. This is particularly true if a large number of clusters are created. Since it is unrealistic to assume that the "true" cluster sizes in every real world data set are similar, we proposed a criterion that does not necessarily produce equally-sized (populated) clusters. In particular, we chose clusters' standard deviations, which indicates dispersions within clusters, as a criterion for bisection. Our rational is that a coherent cluster is densely packed around its centroid and thus has a smaller standard deviation. An incoherent or sparse "cluster," on the other hand, consists of multiple sub-clusters and has a larger standard deviation. Although, both sum of squared error (SSE) and standard deviation may provide information on the dispersion or sparseness of a cluster, while SSE does not reflect the size of clusters (i.e., a larger cluster tends to have a larger SSE), the standard deviations takes the cluster sizes into account. Thus the standard deviation provides information on "relative" dispersion within clusters. By incorporating this criterion, clusters of various sizes that are well concentrated around their centroids would be created.

The standard deviation of cluster $k$ is obtained by the following equation:

$$s_k = \sqrt{\frac{\sum_{i \in c_k} d(x_i, c_k)^2}{N_k}} \tag{1}$$

where $d(x, c_i)$ is the distance between data point $x_i$ (in cluster $k$) and the centroid of cluster $k$ and $N_k$ is the total number of data in cluster $k$.

### 2.3 The Criterion for Termination

To resolve the complications of determining the number of clusters to be created a priori, we have proposed an automated approach to terminate the algorithm when an optimal number of clusters have been created. For this process, we used the decrease rate of the sum of squared error (SSE) as a criterion for terminating the algorithm. The SSE is given as follows:

$$SSE = \sum_{k=1}^{K} \sum_{i \in c_k} d(x_i, c_k)^2 \tag{2}$$

As the number of clusters increases, the SSE would decrease due to the fact that each cluster would consist of a smaller number of, usually coherent, data. However, we expected that there is a point at which an additional cluster would only provide marginal decrement in SSE. We operationally defined that the number of clusters created at that point as the "optimal" number of clusters and then terminated the bisection process at this point.

In more detail, as shown in Eq. 3, the rate of decrease in error was calculated by taking the difference of the current error (time $t$) with the error of the clusters produced one iteration before ($t - 1$), and then dividing the difference with the total error (i.e., initial SSE where there is only one cluster).

$$error_{rate} = \frac{\sqrt{SSE(t-1)} - \sqrt{SSE(t)}}{\sqrt{SSE(0)}} \tag{3}$$

When the resulting quotient becomes less than the termination parameter $\gamma$, then the algorithm is terminated.

### 2.4 Latent Semantic Analysis

As with most text clustering and classification tasks, there are always problems associated with handling extremely high dimensional data sets. To address this problem, we added a feature reduction step, namely the Latent Semantic Analysis (LSA [3]), to our bisecting k-means algorithm. LSA reduces the number of dimensions by identifying "effective" components or dimensions, and we used only these "effective" dimensions for actual clustering. While LSA is usually performed as data preprocessing in typical text mining tasks(e.g. [3]), we performed LSA each time new clusters were created in order to obtain more representative components or feature dimensions for each cluster.

**Table 1.** This table compares the differences between our augmented bisecting k-means algorithm (ABKM) and the standard k-means (KM) and bisecting k-means algorithms (BKM)

| | KM | BKM | ABKM |
|---|---|---|---|
| Requires K a priori | yes | yes | no |
| Bootstrapping | no | no | yes |
| Bisecting Criterion | N/A | cluster size or SSE | Standard Deviation |

# 3 Experiments

In order to evaluate our new augmented bisecting k-means algorithm we performed clustering experiments with the Reuters21578 corpus data. In Experiment 1, we compared the results of standard k-means and bisecting k-means method with our augmented bisecting k-means algorithm. We also performed separate experiments to test the effects of bootstrapping aggregation (Experiment 2) and of the Latent Semantic Analysis (Experiment 3).

## 3.1 The Data

From the Reuters21578 corpus, we created a total of three experimental data sets to examine the performance of each algorithm. For each set of data, we split it into a training set and a test set. In the first training set, we extracted articles of four topics; "trade", "crude oil", "acquisitions," and "earnings," using 200, 200 ,300, and 300 articles respectively. The second training set contained 100 articles on "interest," 200 "trade," 200 "crude oil," 300 "acquisition," and 300 "earnings". The third set contained 100 articles on "money-fx", 100 "interest", 200 "trade", 200 "crude oil," 300 "acquisition," and 300 "earnings". The test sets contained 100 novel articles from each topic used in the training sets. These three data sets were created to examine the effects of the clusters size and the numbers of clusters on the three different clustering algorithms.

## 3.2 Pre-processing the Data

First, we removed any unneeded characters such as punctuation marks and numbers from the articles. We also converted all upper case letters to lower cases. Along with the general stopwords such as "a", "the", "and", we also removed words such as "reported," "told," "according," which are words that occurred in high frequency in news articles. We also used the list of names and places that came with the Reuters data to remove those unneeded terms. We, then, removed 99% sparse terms to reduce the dimensionality of the data (deeming that if a term did not appear in more than 1% of the documents then it was not a needed term for clustering). After removing unneeded terms we performed stemming, which left us with a total of 2341 terms. As for document representation, we used a popular bag of words model with the tf-idf weighting scheme to create a document term matrix for our analyses. The document term matrix was created using R programming language with the text mining package [4].

## 3.3 Experiments 1

In Experiment 1, we used three different stop criterion parameters that ranged from 0.001–0.003, and used the number of clusters produced by our augmented bisecting k-means as the number of cluster to be created (i.e., $k$) for the standard k-means and bisecting k-means algorithms. The training set was used to find centroids by the three different methods. After training, the test sets were used

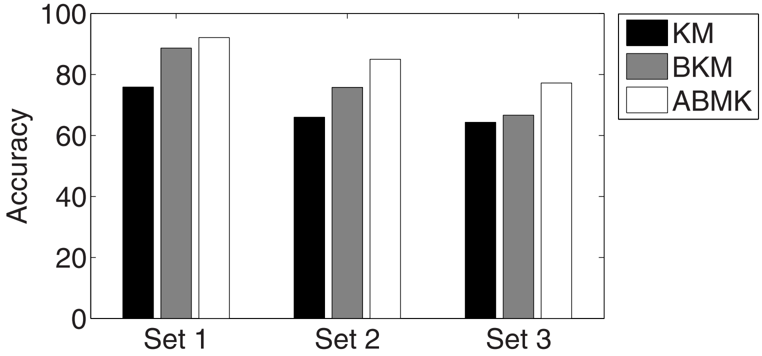


**Fig. 1.** The average classification accuracies for three different clustering algorithms, namely augmented bisecting k-means algorithm (ABKM) and the standard k-means (KM) and bisecting k-means algorithms (BKM)

to classify the articles according to the centroids found from the training sets. Then we calculated the classification accuracies of the test sets to evaluate the three clustering algorithms.

**Results of Experiment 1.** The average classification accuracies of the first experiment are shown in Fig. 1. The average classification accuracy of our novel augmented algorithm (ABKM) for Set 1 was about 15% higher than the original k-means algorithm (KM) and about 4% higher than the standard bisecting k-means algorithm (BKM). Looking at the results of the other two data sets, we were able to see even larger differences in accuracies between our algorithm and BKM. ABKM improved approximately 15% and 20% in classification accuracies compared to BKM and KM, respectively.

The advantage of ABKM over BKM was more noticeable in the experiments with Set 2 and Set 3 where the sizes of the "true" clusters were dissimilar (i.e., the clusters consisted of 100, 200, or 300 articles). In contrast, ABKM performed only marginally better than BKM in Set 1 where the sizes of clusters were similar. This tendency might be attributed to the differences in bisection criteria applied to ABKM (i.e., standard deviation) and BKM (i.e., cluster size). That is, by bisecting a less coherent or sparse clusters, ABKM was able to achieve more accurate clustering than BKM when the "true" cluster sizes varied.

### 3.4 Experiment 2

We examined the effects of the bootstrap aggregating in Experiment 2 by integrating the process into both KM and BKM and excluding the process from ABKM. Set 1 of Experiment 1 was used in the present experiment. The results of Experiment 2 is shown in Table 2. In all k-means algorithms, we observed some

**Table 2.** Result of Experiment 2. This tables shows the effects of adding a bootstrap aggregating procedure to the standard k-means (KM), bisecting k-means (BKM), and augmented bisecting k-means (ABKM).

| Bootstrap Aggregating | KM | BKM | ABKM |
|---|---|---|---|
| Not Incorporated | 75.58% | 87.67% | 90.33% |
| Incorporated | 81.52% | 89.83% | 92.08% |

improvement in classification accuracies, suggesting that the bootstrapping procedure indeed produced more reliable centroids.

### 3.5 Experiment 3

In Experiment 3 we examined the effects of LSA by manipulating the proportions of variance retained by LSA. We performed ABKM clustering where LSA retained 95,75,60, 50 or 30% of the variance. We also tested without performing the LSA process entirely. The results of Exepriment 3 were rather striking: the classification accuracies for ABKM were higher than 92% in all conditions. That is, only 30% of variance of the original data was needed for accurate clustering for the present task, showing no adverse effect of dimension reduction on clustering. Reducing the retention rate significantly decrease the amount of time required for computation without decreasing accuracies. Note again, we performed LSA every time new clusters were created. A similar outcome may not be obtained if one performs LSA only once at the beginning of analysis.

## 4 Discussion and Conclusion

With three different text clustering experiments, we have shown that our algorithm (ABKM) was able to outpreform and produce approximately 10% better classification accuracies than the standard bisecting k-means (BKM) and 15% better than the standard k-means algorithm (KM). The results of the bootstrapping experiments showed that it has positive effect on clustering. However, the difference in classification accuracies between BKM and ABKM might be caused by the change in bisecting criteria. We need more extensive experiments and analyses in order to confirm this speculation. Lastly, we were also able to show that by using LSA we could cut the calculation time without adversely affecting the precision of clustering.

We were able to show some advantages of using our augmented algorithm from these experiments, but there are some issues associated with our algorithm. Although LSA significantly reduces computational time, ABKM still require more intensive computation than BKM and KM. It probably needs some algorithm to terminate the bootstrapping process when additional resampling does not provide any further information (for determining accurate centroids) in order to prevent from running unnecessary analyses.

The other issue is inherent to any bisecting k-means clustering. Once a cluster is bisected into two cluster, the clusters that have been separated will never end up back in the same cluster. This becomes a problem because the first few bisections are often coarse (there are still a large number of data which belong in different topics) and the likelihood of documents of the same topic being separated can be high. This leads to documents that are of the same topic but in different clusters. To resolve this problem we are currently extending our algorithm to integrate a process of merging clusters.

We have only tested and compared ABKM against the standard versions of the k-means and bisecting k-means algorithms, using a relatively small number of topics and articles. In order to have a more comprehensive evaluation we should compare our algorithm against methods that are growing in popularity such as the self organizing maps [1] or algorithms that make use of evolutionary strategies with a variety of data sets

Nonetheless, by integrating (1) a bootstrap aggregating procedure, (2) a bisecting criteria that relies on dispersions of data within clusters, and (3) a method to automatically terminate the algorithm when an optimal number of clusters have been produced, we were able to improve the original bisecting k-means algorithm. The improvement was more apparent when the sizes of "true" clusters were dissimilar.

## Acknowledgements

This work was in part supported by the Japan Society for the Promotion of Science KAKENHI (Grant No. 20700235) and Support Center for Advanced Telecommunications Technology Research (SCAT).

## References

1. Dittenbach, M., Rauber, A., Merkl, D.: Uncovering Hierarchical Structure in Data Using the Growing Hierarchical Self-organizing Map. Neurocomputing 48, 199–216 (2002)
2. Steinbach, M., Karypis, G., Kumar, V.: A Comparison of Document Clustering Techniques. In: Proc. of Workshop on Text Mining, 6th ACM SIGKDD Int. Conf. on Data Mining (2000)
3. Torkkola, K.: Discriminative Features for Text Document Classification. Pattern Analysis Application 6, 301–308 (2003)
4. Feinerer, I., Hornik, K., Meyer, D.: Test Mining Infrastructure in R. Journal of Statistical Software 25 (2008)

# Ontology Based Personalized Modeling for Type 2 Diabetes Risk Analysis: An Integrated Approach

Anju Verma[1], Maurizio Fiasché [2,3], Maria Cuzzola[3], Pasquale Iacopino[3], Francesco C. Morabito[2], and Nikola Kasabov[1]

[1] Knowledge Engineering and Discovery Research Institute,
Auckland University of Technology
Auckland
anju.verma@aut.ac.nz
[2] DIMET, University Mediterranea of Reggio Calabria, Italy
[3] Transplant Regional Center of Stem Cells and Cellular Therapy, "A. Neri",
Reggio Calabria, Italy
maurizio.fiasche@unirc.it

**Abstract.** A novel ontology based type 2 diabetes risk analysis system framework is described, which allows the creation of global knowledge representation (ontology) and personalized modeling for a decision support system. A computerized model focusing on organizing knowledge related to three chronic diseases and genes has been developed in an ontological representation that is able to identify interrelationships for the ontology-based personalized risk evaluation for chronic diseases. The personalized modeling is a process of model creation for a single person, based on their personal data and the information available in the ontology. A transductive neuro-fuzzy inference system with weighted data normalization is used to evaluate personalized risk for chronic disease. This approach aims to provide support for further discovery through the integration of the ontological representation to build an expert system in order to pinpoint genes of interest and relevant diet components.

**Keywords:** disease ontology, integrated approach, personalized modeling, diabetes.

## 1 Introduction

Populations are aging and the prevalence of chronic diseases which persists for many years is increasing. The chronic diseases such as cardiovascular disease, type 2 diabetes and obesity have high global prevalence, have multifactorial aetiology and are mainly caused by interactions of a number of common factors including genes, nutrition and life-style. For ontology based personalized risk evaluation for type 2 diabetes, a Protégé-based ontology has been developed for entering data for type 2 diabetes and linking and building relationships among concepts. The ontological representation provides the framework into which information on individual patients for disease symptoms, gene maps, diet and life history details can be inputted, and risks, profiles, and recommendations derived. A personalized risk evaluation system has been used

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 360–366, 2009.

for building the personalized model. Global models capture trends in data that are valid for the whole problem space, and local models capture local patterns which are valid for clusters of data. Both models contain useful information and knowledge. Local models are also adaptive to new data as new clusters and new functions that capture patterns of data in these clusters. A local model can be incrementally created. Usually, both global and local modeling approaches assume a fixed set of variables and if new variables, along with new data, are introduced with time, the models are very difficult to modify in order to accommodate these new variables. However new variables be accommodated in personalized models, as they are created ''on the fly'' provided that there is data for them [1]. The next two sections will describe a chronic disease ontology for type-2 diabetes and personalized risk evaluation method (TWNFI) which is used to build personalized models , its comparison with global and local models and the last section of the paper describes the framework for integration of chronic disease ontology and personalized modeling.

## 2 Chronic Disease Ontology

Ontology is a systematic account of being or existence. Ontology in terms of bioinformatics can be interpreted as the representation of the existing domain of the knowledge of life. Ontology is used to reason and make inferences about the objects within the domain [2]. Ontology is concerned with making information and knowledge explicit; it includes descriptions of concepts and their relationships. Ontology describes a hierarchical structure of concepts and the relationships built in order to extract new knowledge. Ontology is generally written as a set of definitions of the formal vocabulary of objects and relationships in the given domain. It supports the sharing and reuse of formally represented knowledge among systems [3, 4]. As a database technology, ontologies are commonly coded as triple stores (subject, relationship, object), where a network of objects is formed by relationship linkages, as a way of storing semantic information [5, 6]. A standardized ontology framework makes data easily available for advanced methods of analysis, including artificial intelligence algorithms, that can tackle the multitude of large and complex datasets by clustering, classification, and rule inference for biomedical and bioinformatics applications. The main advantages of building ontology are to extract and collect knowledge; share knowledge; manage terminology; store, retrieve and analyze; find relationships between the concepts; discover new knowledge and reuse knowledge for decision support system. Chronic disease ontology consists of five major domains namely; organism domain, molecular domain, medical domain, nutritional domain and a biomedical informatics map domain. These domains or classifications contain further subclasses and instances. Each subclass has a set of slots which provide information about each instance and have relationships among other slots, instances and concepts. Each gene instance has different information associated with the gene and also has relationships with other domains. The advantage of this chronic disease ontology is that it can be updated manually and regularly with new knowledge and information providing a framework to keep an individual's specific information (medical, genetic, clinical and nutritional), to discover new knowledge and to adapt as required for personalized risk prediction and advice.

## 3 Personalized Risk Evaluation System for Type 2 Diabetes

Type 2 diabetes mellitus is one of the most common chronic “lifestyle” diseases with a high prevalence throughout the world [7]. There are two main types of diabetes mellitus; type-1 and type-2. Type 2 diabetes is the most common type of diabetes and globally about 90% of all cases of diabetes are type 2 diabetes [8]. There have been several models, namely, ‘The global diabetes model’ [9, 10], ‘The diabetes risk score’ [11], the ‘Archimedes diabetes model’ [12, 13], the Diabetes risk score in Oman [14], the ‘Genetic Risk Score’ [15]. All these models predict risk of future complications associated with type 2 diabetes in people with diagnosed type 2 diabetes. The global diabetes model (GDM) is a continuous, stochastic micro simulation (individual by individual approach) model of type 2 diabetes. The GDM is a computer program and predicts longevity, quality of life, medical events and expenditures for groups and individuals with type 2 diabetes. The GDM calculates rates and probabilities of the medical events in diabetic individuals [9, 10]. It has been reported that from the existing methods for predicting risk of type 2 diabetes, The Archimedes Model predicts the risk with better sensitivity and specificity than other models [16]. Recently, the ‘Genetic Risk Score’ has been developed which uses multiple genetic as well as conventional risk factors [15]. Because these methods calculate risk of type 2 diabetes globally and they are not the same as the proposed methodology in this thesis, which involves calculations of personalized risk. The aim of the current research is to create a personalized model for predicting risk of type 2 diabetes. Genetic variables have been used along with clinical variables to create a personalized model to predict risk of type 2 diabetes. The next section of this paper describes the data and methods used for creating a diabetes risk model using genetic markers along with clinical variables.

### 3.1 Feature Selection for Building the Personalized Risk Evaluation System for Type-2 Diabetes

The first step to build the model was feature selection which has been done by using different methods including signal to noise ratio and t-test. This analysis was done using NeuCom and Siftware. Results achieved from signal to noise ratio are exactly similar to student’s t-test. According to signal to noise ratio and t-test for the combined male and female subjects, genes ANGPTL3, ANGPT4, TNF, FLT1, MMP2 and CHGA are ranked highest. Interestingly, gene CHGA has not been ranked at same high position for male and female subjects separately. The first six genes of highest importance for male and female subjects were selected for further analysis and to build personalized sex-specific risk prediction model. As genes are ranked differently as per signal to noise ratio for male and female subjects, different genes have been selected for personalized modeling. Different methods were then used for type 2 diabetes risk prediction methods such as multiple linear regression using NeuCom (global, inductive method), WWKNN and TWNFI (personalized methods) [17].

### 3.2 Examples of Personalized Modeling

As every person has a different genetic admixture, therefore personalized prediction and treatment is required for each person. Table 1 shows results from example of

**Table 1.** Examples of TWNFI personalized models for two different male subjects; high risk and low risk; with weight of variables and genes with global weights representing importance of the variables

| | Subject 1 (High risk male) | | Subject 2 (Low risk male) | | |
|---|---|---|---|---|---|
| Input Variables | Values of input | Weights of input variables | Values of input | Weights of input variables | Global weights/ importance (male) |
| Age (years) | 58 | 0.7729 | 42 | 0.9625 | 0.8393 |
| Haemoglobin (g/L) | 12 | 0.8521 | 15.4 | 0.7847 | 0.8429 |
| Fasting blood glucose (mmol/L) | 5.6111 | 0.7507 | 5.1111 | 0.9352 | 0.8769 |
| Cholesterol (mmol/L) | 4.7582 | 0.7478 | 5.3013 | 0.752 | 0.8104 |
| Triglycerides (mmol/L) | 2.1225 | 0.6961 | 2.6983 | 0.7413 | 0.8327 |
| ANGPTL3 | 32.14 | 0.7617 | 1 | 0.9269 | 0.9254 |
| FGF1 | 17.9446 | 0.7295 | 1 | 0.641 | 0.8228 |
| FLT1 | 24.059 | 0.651 | 1 | 0.7059 | 0.8096 |
| MMP2 | 4.4584 | 0.6797 | 1 | 0.8802 | 0.9009 |
| TNF | 3.3048 | 1 | 1 | 0.8495 | 0.8699 |
| ANGPT4 | 14.1165 | 0.6705 | 1 | 1 | 0.904 |
| Actual output | 1 | | 0 | | |
| Predicted output with Multiple linear regression | | 0.7963 | | 0.1378 | |
| Predicted output with WWKNN | | 1.127 | | 0 | |
| Predicted output with TWNFI | | 1.002 | | 0 | |

personalized model built for two male subjects. Subject 1 belongs to class 1(with type 2 diabetes) and subject 2 belongs to class0 (without type 2 diabetes).

It was found that highest accuracy was achieved with the TWNFI method. TWNFI not only gives highest accuracy, also gives weights of variables as per their importance for risk of disease. For each subject in present example, separate weight of each variable has been presented and compared with global weights of variables for male subjects. It is very interesting that male subject 1 and 2 both have higher values of fasting blood glucose, cholesterol and triglycerides, the genes were more important factors to predict the risk of type 2 diabetes for male subject 2. By comparing weights for each variable of each subject, it was found that for male subject 1, gene TNF was found to be the most important gene associated with type 2 diabetes while for male subject 2, ANGPT4 gene has been weighted the highest, while for all the male subjects the ANGPTL3 gene has been found most important factor for type 2 diabetes. TWNFI along with high accuracy and importance of variables also provides set of rules based on the clusters formed based on nearest neighbors. Each rule contains a lot of information. Rules or profiles for male subjects were generated on the basis of nearest samples.

# 4 Integration of Chronic Disease Ontology and Personalized Modeling

The chronic disease ontology that was described in section 2 will be used to integrate personalized modeling and ontology. The current chronic disease ontology contains most of genes which are common for three chronic interrelated diseases (cardiovascular disease, type-2 diabetes and obesity).

Data was collected during a Government Research Program with the title: ”Biologia e Impiego dei Progenitori Endoteliali nell’ Arteriopatia Obliterante Periferica” sponsored from Italian Ministry of the Health. In the present Italian dataset used for predicting risk of type 2 diabetes clinical and genetic variables were included and it was found that for male and female subjects different combinations of genes are more predictive of type 2 diabetes. So these genes were updated in the chronic disease ontology and the missing genes and information related to these genes was also added in to the chronic disease ontology. Similarly, any other information derived from personalized model can be added to the chronic disease ontology and the new relationships and discoveries within the chronic disease ontology can be used to improve personalized risk evaluation system. The framework uses the chronic disease ontology based data and knowledge embedded in the ontology. It also allows the adaptation of new knowledge by entering the results of the machine learning system to ontology. The first module for integration is protégé based the chronic disease ontology which is knowledge and data repository module, second module is TWNFI; a personalized modeling technique and an interface between both the modules.

## 4.1 Example for Integration of the Chronic Disease Ontology and Personalized Diabetes Risk Analysis Model

Integration of the chronic disease ontology and personalized model can be done for diabetes. For example, the information obtained from personalized model for type 2 diabetes in the Italian dataset, such as the gene matrix metalloproteinase (MMP2), responsible for protein binding in normal person and mutated form is responsible for high risk of type 2 diabetes in the Italian male population can be added to the ontology and if a new male subject comes which is from Italian population, the same information can be used next time. Similarly, it has been found that the gene hypoxia inducible factor 1 (HIF1A) acts as a normal transcription binding factor but mutation in gene is related to type 2 diabetes in females in Italian population. This information can be added to ontology and can be applied to the analysis for the next new subject from a similar population and with similar clinical features for risk prediction. Similar process can be applied for predicting risk of obesity. Recently, it was found that gene FTO in its inactivated state protects from risk of obesity [18]. Polymorphism in FTO gene is strongly and positively correlated to body mass index which is common measure of obesity. This knowledge has been updated in the chronic disease ontology and the system is able to use this knowledge if a similar subject with high body mass index comes, it can identify that FTO gene is active and the person may have a predisposition to obesity if dietary intake exceeds physical activity by upsetting.

## 5 Conclusions

Diabetes has global prevalence and there have been many methods developed to predict risk of type 2 diabetes. All the methods developed so far use general and clinical variables only. None of the methods so far published have combined clinical and genetic variables together. This chapter describes how a model was built using clinical and genetic variables. In particular, it has been found that

- Male subjects have high values of cholesterol and triglycerides and are more prone to type 2 diabetes.
- For male and female subjects different combinations of genes have association with type 2 diabetes.
- For male subjects, genes ANGPTL3, MMP2, ANGPT4, TNF, FGF1 and FLT1 appear to be the most important genes associated with risk of type 2 diabetes.
- For female subjects, genes ANGPTL3, ANGPTL4. HIF1A, TNSF15, FLT1 and TNF appear to be the most important factors for determining risk of type 2 diabetes.

For personalized modeling, different methods such as WWKNN and TWNFI were used and compared. It has been found that TWNFI gives highest accuracy along with importance of each gene and variable by optimizing each variable and weight which can be used for better prediction and recommendations. The personalized risk evaluation system explained in this paper utilizes 6 genes at this stage and it can be further extended with more genes and more set of clinical and general variables. Still a better prediction system can be developed, if nutritional information and other environmental variables are known (e.g. exposure to sun for vitamin D) along with clinical and genetic variables are available.

**Acknowledgments.** This study has been supported by Foundation of Research and Technology by TIF scholarship through an affiliate of Pacific Channel Limited and KEDRI, AUT. We would like to thank Ministry of Health for providing Health related data (NNS97). The data has been provided by the Transplant Regional Center of Stem Cells and Cellular Therapy, ”A. Neri” of Reggio Calabria, Italy. Many thanks to Maurizio Campolo of University of Reggio Calabria for his aid in our research activity.

## References

1. Kasabov, N.: Global, local and personalized modeling and profile discovery in Bioinformatics: An integrated approach. Pattern Recognition Letters 28(6), 673–685 (2007)
2. Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition 5, 199–220 (1993)
3. Fensel, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, 2nd edn. Springer, Heidelberg (2004)
4. Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What are ontologies, and why do we need them? Intelligent Systems and Their Applications 14, 20–26 (1999)

5. Owens, A.: Semantic Storage: Overview and Assessment. Technical Report IRP Report 2005, Electronics and Computer Science, U of Southampton (2005)
6. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, May 17 (2001)
7. The FIELD Study Investigators, The need for a large-scale trial of fibrate therapy in diabetes: the rationale and design of the Fenofibrate Intervention and Event Lowering in Diabetes (FIELD) study. ISRCTN64783481. Cardiovascular Diabetology 2004, 3, 9 (2004)
8. New Zealand Guidelines Group. Management of diabetes. New Zealand Guidelines Group, Wellington (2003(a)) (Retrived), `http://www.nzgg.org.nz/guidelines/dsp_guideline_popup.cfm?guidelineID=36`
9. Brown, J.B., Palmer, A.J., et al.: The Mt. Hood challenge: cross-testing two diabetes simulation models. Diabetes Research and Clinical Practice 50(3), S57–S64 (2000(a))
10. Brown, J.B., Russell, A., et al.: The global diabetes model: user friendly version 3.0. Diabetes Research and Clinical Practice 50(3), S15–S46 (2000(b))
11. Lindstrom, J., Tuomilehto, J.: The diabetes risk score. A practical tool to predict type-2 diabetes risk. Diabetes Care 26(3), 725–731 (2003)
12. Eddy, D.M., Schlessinger, L.: Archimedes. A trial-validated model of diabetes. Diabetes Care 26(11), 3093–3101 (2003(a))
13. Eddy, D.M., Schlessinger, L.: Validation of the Archimedes diabetes model. Diabetes Care 26(11), 3102–3110 (2003(b))
14. Al-Lawati, J.A., Tuomilehto, J.: Diabetes risk score in Oman: A tool to identify prevalent type-2 diabetes among Arabs of the Middle East. Diabetes Research and Clinical Practice 77, 438–444 (2007)
15. Cornelis, M., Qi, L., et al.: Joint effects of common genetic variants on the risk of type-2 diabetes in U. S. men and woemen of European ancestry. Annals of Internal Medicine 150, 541–550 (2009)
16. Stern, M., Williams, K., et al.: Validation of prediction of diabetes by the Archimedes Model and comparison with other predictiong models. Diabetes Care 31(8), 1670–1671 (2008)
17. Song, Q., Kasabov, N.: TWNFI - a transductive neuro-fuzzy inference system with weighted data normalization for personalized modeling. Neural Networks 19(10), 1591–1596 (2006)
18. Fischer, J., Koch, L., et al.: Inactivation of the Fto gene protects from obesity. Nature 458, 894–899 (2009)

# A Pulse-Coupled Network of SOM

Kai Kinoshita and Hiroyuki Torikai

Dept. of Systems Innovation, Grad. school of Engineering Science, Osaka University
1-3 Machikaneyama-cho, Toyonaka, Osaka 560-8531

**Abstract.** In this paper, we present a resonate-and-fire type spiking neuron model and analyze its stability of synchronization phenomena to an input spike train. We also present a parameter update rule to memorize a period of input spike train and analyze its pull-in process. Then we present a pulse-coupled network of the proposed model and show that the proposed network and a Kohonen's SOM has similar functions. And we make comparisons to a network of another resonate-and-fire type neuron model and show superiority of the proposed model.

## 1 Introduction

Many spiking neuron models have been presented and investigated [1,2,3,4,5,6,7]. For example, resonate-and-fire models have been used to investigate spike-based information coding and processing functions and have been implemented on electric circuits [2,4,5,6,7]. Inspired by such resonate-and-fire models, we have proposed a subthreshold oscillating neuron (SO) model that can be regarded as a piecewise linearised version of simple Izhikevich model or a generalized version of Mitsubori-Saito (MS) model [8]. Also, resonate-and-fire models have been used to construct pulse-coupled neural networks (PCNNs) whose application potential include image processing based on synchronization phenomena [9,10,11]. For example, a self-organizing map (SOM) using digital phase-locked loops (DPLLs), which can be regarded as a resonate-and-fire models, was presented [12,13]. Inspired by such DPLL SOM, we have proposed a self-organizing network of spiking neurons (SSN) that can be regarded as a pulse-coupled network version of an SOM.

In this paper, first, we propose the SO model and analyze its stability of synchronization phenomena to an input spike train. Next, we present an update rule of a dynamic parameter and analyze its pull-in process. We then show that the SO model can memorize input period. Third, we present a pulse-coupled network of SO models and show that the network can have an SOM-like clustering function. Forth, we make comparisons to Kohonen's basic SOM (BSOM) and a pulse-coupled network of MS models as shown in Table: 1: (1)the SSN of the SO model has similar processes to those of the BSOM, (2)the operation of the BSOM uses numerical calculation whereas the operation of the presented model uses nonlinear dynamics, and (3)the SSN of the MS model doesn't have an SOM-like function although return maps and bifurcation diagrams of those models are fitted.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 367–375, 2009.

We emphasize that typical implementations of an SOM need a minimum value detector for implementation of the winner-take-all (WTA) process but the SSN doesn't because of replacing the WTA process by competitive process. We also emphasize that the SSN memorizes input pulse train intervals to a natural frequency of the SO model without a phase detector unlike [13]. In [8], we have synthesized the SO model and an SOM to prototype the SSN. However, in that paper, stability analysis of the SO model, discussion about the basic SOM function and comparison to other pulse-coupled networks were insufficient.

## 2 A Spiking Neuron Model

In this section, we introduce a new spiking neuron. It is a nonlinear subthreshold oscillating neuron model which is easily implemented by an electronic circuit. We use the model as a network element in the next section. Let $t \in \mathbb{R}_+ = \{t|t \in \mathbb{R}, t \geq 0\}$ be a continuous time. Then we define a periodic input $u(t)$ whose period is $T \in \mathbb{R}_+$ i.e., $u(t) := \sum_{n=1}^{\infty} \delta(t - nT)$ where $\delta$ is the Dirac's delta function. We use two state variables $v$, $r\in \mathbb{R}$ and one weight parameter $w\in \mathbb{R}$. $v$ corresponds to a membrane potential and $r$ corresponds to a recovery variable of Izhikevich's simple model (2003) respectivety, and $w$ corresponds to a synaptic weight. For simplicity, we use a state vector $X = [v, r]^T\in \mathbb{R}^2$ and a weight vector $W = [w, 0]^T\in \mathbb{R}^2$. The subthreshold dynamics of a subthreshold oscillating spiking neuron (SO) model is described by the following equation.

$$\dot{X} = AX + B + Wu$$
$$\begin{cases} A = \begin{bmatrix} -1 & -b \\ b & -1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ for } v < \eta \\ \\ A = \begin{bmatrix} 1 & -b \\ b & -1 \end{bmatrix}, B = \begin{bmatrix} 2\eta \\ 0 \end{bmatrix} \text{ for } v > \eta \end{cases} \tag{1}$$

First, we consider an autonomous and subthreshold oscillating case, i.e., $W = [0, 0]^T$ and $v$ stays less than a firing threshold. We use two parameters $b, \eta\in \mathbb{R}$ to characterize the subthreshold behavior. In this paper, we fix $\eta = -20$. So the control parameter is $b$. Fig.1 shows waveforms and phase space trajectories. We can see that the SO model oscillates periodically. We can obtain the approximate period $P \in \mathbb{R}$ by the following SO model's eigen values $\lambda$. $\lambda = 1 \pm bi$ for $v < \eta$, $\lambda = \pm bi\sqrt{1 - \left(\frac{1}{b}\right)^2}$ for $\eta < v$. We assume that $|b|$ is much larger than 1, i.e., $(1/b)^2 \simeq 0$. Then, the natural period $P$ is approximated by the parameter $b$, i.e., $P \simeq 2\pi/b$.

Secondly, we consider an autonomous and firing case, i.e., $W = [0, 0]^T$ and $v$ is greater or equal to the firing threshold $\theta \in \mathbb{R}$. Let $t_n\in \mathbb{R}_+$ be the $n$-th moment when the state $v$ reaches the firing threshold $\theta$, i.e., $t_1 = min\{t|v(t) \geq \theta, t \geq 0\}$, $t_{n+1} = min\{t|v(t) \geq \theta, t > t_n\}$. Let $t^+ = \lim_{\epsilon \to +0} t + \epsilon$. At the moment $t = t_n$, the state vector $X$ is reset to $[c, r]^T$, i.e., $X(t^+) = \begin{bmatrix} c \\ r(t) \end{bmatrix}$ if $v(t) \geq \theta$ where $c\in \mathbb{R}$ is

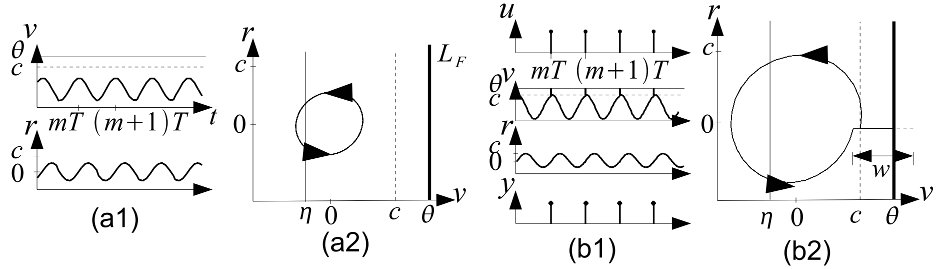


**Fig. 1.** Typical behaviors of the SO models. (a1) shows time series of the states , input and output in autonomous case. (a2) shows its trajectories. (b1) shows time series of the states , input and output in non-autonomous case. (b2) shows its trajectories. ($b$, $\eta$, $\theta$, $c$) = (10, −20, 60, 40). Input period $T = 2\pi/10$.

a reset parameter. We refer to such a reset as a self-firing. An output of the SO model is given by $y(t) := \sum_{n=1}^{\infty} \delta(t - t_n)$.

Third, we consider a non-autonomous case, i.e., $W \neq [0, 0]^T$. We define $t^- = \lim_{\epsilon \to -0} t + \epsilon$ and $t^{++} = \lim_{\epsilon \to +0} t^+ + \epsilon$. At the moment $t = nT$, the state vector $X$ jumps to $X + W$ by an input pulse $u(nT)$, i.e., $X(t^+) = X(t^-) + W$ if $t = nT$. If the state $v(t^+)$ overs the firing threshold $\theta$, the state vector $X$ is reset to $[c, r]^T$, i.e., $X(t^{++}) = \begin{bmatrix} c \\ r(t^-) \end{bmatrix}$ if $v(t^+) \geq \theta$. We refer to such a reset as a compulsory-firing. The exact solution of the state $X(t)$ can be obtained by solving the piecewise linear equation in Eq(1).

We analyze the behavior of discrete-time state $r[n] \in \mathbb{I} \equiv \{r(t)|t = nT\}$ in the compulsory-firing case using a return map (Poincare map). We define the following Poincare section $L_F$.

$$L_F = \{(v, r, t)|v = \theta, t = nT\} \tag{2}$$

We assume that the weight $w$ is large enough, e.g., $w = 100$. Then, we can obtain a return map $F$, i.e.,

$$F : L_F \to L_F, \quad r[n] \mapsto r[n+1] \tag{3}$$

An analytical formula of the return map $F$ is obtained by solving the piecewise linear equation in Eq(1). Fig.2 shows the return map $F$. In Fig.2, there are attractors $A$, $B$ and $C$ in (a), (b) and (c) respectively. We can see that each attractor $A$, $B$ and $C$ in Fig.2 is a fixed point or a periodic attractor. In Fig.2, the state $r[n]$ with a periodic input $u(t)$ is stable when the weight $w$ is large enough. Consequently, we can say that the analitical result of the state $r[n]$ is stable in compulsory-firing case.

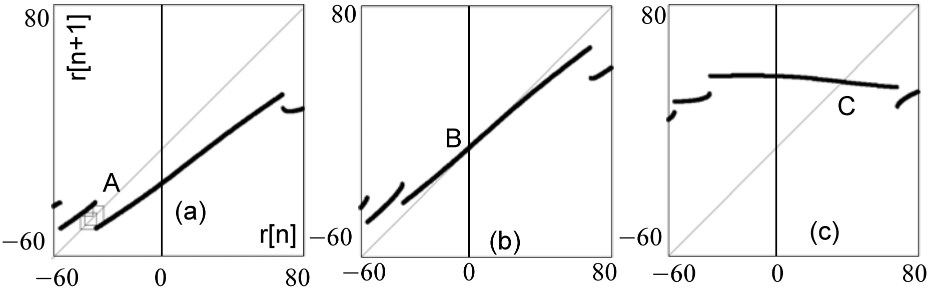


**Fig. 2.** Return maps of the SO model whose poincare section is $L_F = \{(v, r)|v = \theta, t = nT\}$. $[\eta,\ \theta,\ c] = [-20,\ 60,\ 40]$. Input period $T = 2\pi/10$. (a)$P = 2\pi/9$, (b)$P = 2\pi/10$, (c)$P = 2\pi/12$.

### 2.1 Parameter Update Rule

We expand the SO model to memorize a period of the input $u(t)$ by updating the parameter $b$. We update the parameter $b$ when a compulsory-fire occurs as the follows.

$$\begin{aligned} &\text{if} \quad t = t_n = mT, \text{ then} \\ &\quad b(t^{++}) = b(t^{-}) + g(r(t^{-})) \\ &\quad g(r) := \begin{cases} \alpha r & \text{for } L_l < r < L_u \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

We refer to such an update for $g(r) \neq 0$ as valid update and refer to the parameter $b$ as a dynamic parameter. Fig.3 shows a bifurcation diagram of the dynamic parameter $b$ for the input period $T$. As shown in Fig.3, the input period $T$

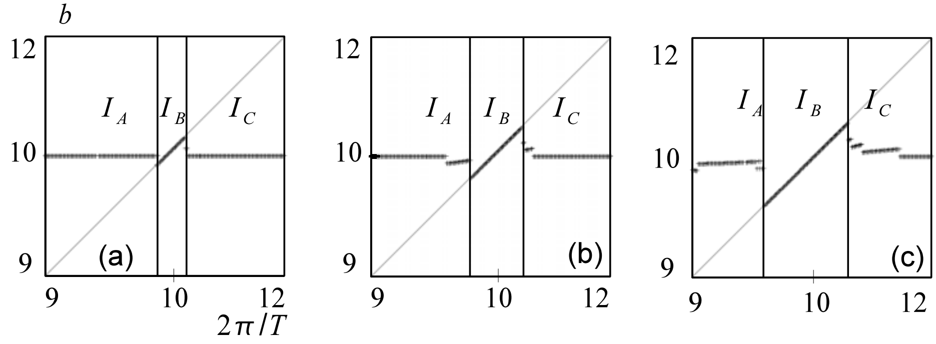


**Fig. 3.** Typical phenomena of pull-in process of the dynamic parameter $b$ for the input period $T$. $[\eta,\ \theta,\ c,\ \alpha,\ L_l,\ L_u] = [-20,\ 60,\ 40,\ -\pi/60,\ -30,\ 30]$. Initial parameter $b_0$ is fixed to 10. (a) shows $w = 15$ case. (b) shows $w = 30$ case. (c) shows $w = 100$ case.

can be split into $I_A$, $I_B$ and $I_C$. From Fig.3, if the input period $T$ is close to $P(0) = 2\pi/b(0)$, then $2\pi/b^* := \lim_{t\to\infty} 2\pi/b(t)$ is near $T$, i.e., $2\pi/b^* \fallingdotseq T$ for $T \in I_B$.

## 3 A Pulse-Coupled Neural Network

In this section, we introduce the self-organizing network of spiking neurons (SSN) interconnected with the spike train. Let $i \in \mathbb{N}$ be an index of each spiking neuron and $t_{n,i} \in \mathbb{R}_+$ be the $n$-th firing moment of the $i$-th neuron. Then we use state variables $v_i, u_i \in \mathbb{R}$ , weight parameter $w_i \in \mathbb{R}$ and dynamic parameter $b_i \in \mathbb{R}$. For simplicity, we use a state vector $X_i = [v_i, u_i]^T \in \mathbb{R}^2$ and a weight vector $W_i = [w_i, 0]^T$. The single neuron is described by the following equations.

$$\dot{X}_i = A_i X_i + B + W_i u$$
$$\begin{cases} A_i = \begin{bmatrix} -1 & -b_i \\ b_i & -1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ for } v_i < \eta \\ \\ A_i = \begin{bmatrix} 1 & -b_i \\ b_i & -1 \end{bmatrix}, B = \begin{bmatrix} 2\eta \\ 0 \end{bmatrix} \text{ for } v_i > \eta \end{cases} \tag{5}$$

At the $n$-th firing moment of the $i$-th neuron i.e., $t = t_{i,n}$, the state vector $X_i$ is reset to $[c, r_i]^T$, i.e., $X(t^+) = \begin{bmatrix} c \\ r_i(t) \end{bmatrix}$ if $t = t_{n,i} \geq \theta$. The output $y_i$ of the $i$-th neuron is given by $y_i(t) := \sum_{n=1}^{\infty} \delta(t - t_{n,i})$. In the SSN, the SO models are connected by the following two processes, competitive process and parameter update process.

1. The competitive process is described by two update rules of weight $w_i$.
   (a) At the $n$-th input moment i.e., $t = nT$, the weight $w_i$ jumps to $w_i + \delta w$, i.e.,

$$\begin{aligned} &\text{if } \ t = nT \text{ then} \\ &\quad w_i(t^+) = w_i(t^-) + \delta w, \text{ for all } i \end{aligned} \tag{6}$$

   (b) At the $n$-th firing moment of the $j$-th neuron i.e., $t = t_{n,j}$, the weight $w_i$ jumps to $w_c \in \mathbb{R}$, i.e.,

$$\begin{aligned} &\text{if } \ t = t_{n,j} \text{ then} \\ &\quad w_i(t^+) = w_c, \text{ for all } i \text{ except } i = j \end{aligned} \tag{7}$$

2. The parameter update process is given by a update rule of dynamic parameter $b_i, b_{i\pm1}$.
   (a) At the moment $t = t_{n,j}$, we update dynamic parameter $b_j, b_{j\pm1}$ when the $j$-th neuron's compulsory-fire occurs, i.e.,

$$\begin{aligned} &\text{if } \ t = t_{n,j} = mT \text{ then} \\ &\quad \begin{cases} b_j(t^{++}) = b_j(t^-) + g(r_j(t^-)) \\ b_{j\pm1}(t^+) = b_{j\pm1}(t^-) + \delta b \end{cases} \end{aligned} \tag{8}$$

(b) In this paper, we set 10 neurons in line $\delta b := \begin{cases} 0.1 & \text{if } b_i - b_{i\pm1} > 0 \\ -0.1 & \text{if } b_i - b_{i\pm1} < 0 \end{cases}$.

Now we simulate the SSN of the SO models. We select input period from $\{T_A := 2\pi/10, T_B := 2\pi/11, T_C := 2\pi/12\}$ randomly. We set 10 neurons in line and initial parameter $b_i(0) = 10$ for all $i$. Fig.4(a) shows time waveforms of the indices of the firing neurons. Fig.4(b) shows time waveforms of the dynamic parameter $b_i$ for all neurons. Fig.4(c) shows the dynamic parameter $b_i$ at the end of the simulation. We can see that the dynamic parameters are clustered into $N_A$, $N_B$ and $N_C$ in Fig.4(c). So, we can say that the SNN learns the input periods by the dynamic parameter $b_i$.

Next, we compare the SSN to a basic self-organizing map (BSOM). The BSOM uses numerical calculations to realize the competitive process (winner-take-all) and the parameter updates. The BSOM is described as a serial of two processes, winner take-all process and reference vector update process. Table:1 shows com-

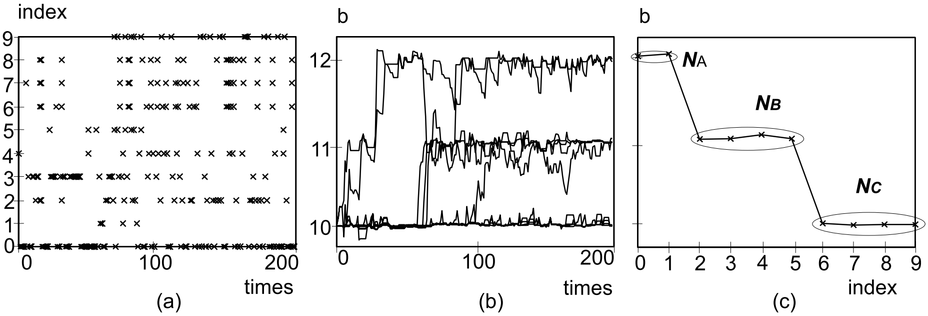


**Fig. 4.** A typical result of the ONLINE simulation of the SSN of the SO model. This SSN is composed of 10 neurons. (a) shows the firing moments $t_{n,i}$ for all $i$ in $t-i$ plane. (b) shows the transition of the dynamic parameter $b_i$ for all $i$ in $i-b$ plane. (c) shows the dynamic parameter $b_i$ in a steady state in $i-b$ plane. $[\eta, \theta, c, \alpha, L_l, L_u, \delta w, w_c] = [-20, -\pi/60, 60, 40, -30, 30, 5, 5]$. Initial dynamic parameter $b_i(0)$ is fixed to 10 for all $i$. Simulation is done for 200 times and the input intervals are selected randomly from $\{T_A = 2\pi/10, T_B = 2\pi/11, T_C = 2\pi/12\}$.

**Table 1.** Comparison among Kohonen's basic SOM, proposed SSN of SO model and SSN of MS model

| Kohonen's BSOM | SSN of SO model | SSN of MS model |
|---|---|---|
| Winner take-all process by numerical calculations | Valid competitive process by nonlinear dynamics | Invalid competitive process by nonlinear dynamics |
| Reference update process by numerical calculations | Parameter update process by nonlinear dynamics | |
| Similar SOM functions as shown in Fig:4 and Fig:5 | | Different from Fig:4 and Fig:5 as shown in Fig:6 |

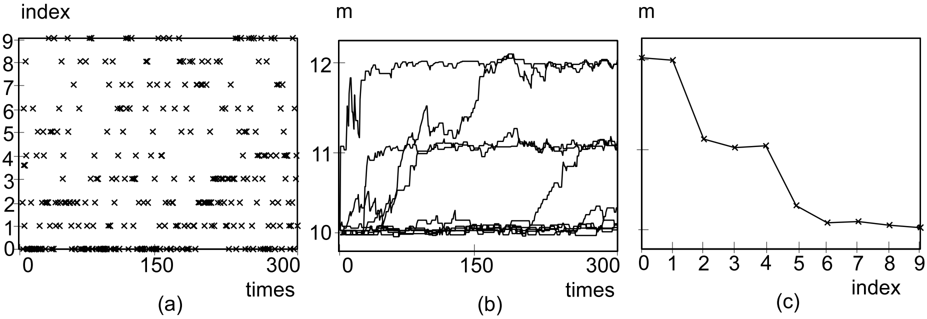


**Fig. 5.** A typical result of the ONLINE simulation of the Kohonen's BSOM. This BSOM is composed of 10 units. (a) shows the winning moments for all $i$ in $t - i$ plane. (b) shows the transition of the reference $m_i$ for all $i$ in $i - m$ plane. (c) shows the reference $m_i$ in a steady state in $i - m$ plane. In this simulation, the reference vector update process is $m_c(t + 1) = m_c(t) + 0.5 \cdot (x(t) - m_c(t))$ and $m_{c\pm1}(t + 1) = m_{c\pm1}(t) + 0.1 \cdot (x(t) - m_c(t))/|x(t) - m_c(t)|$. Initial reference $m_i(0)$ is fixed to 10 for all $i$. Simulation is done for 300 times and the input variables are selected randomly from $\{10, 11, 12\}$.

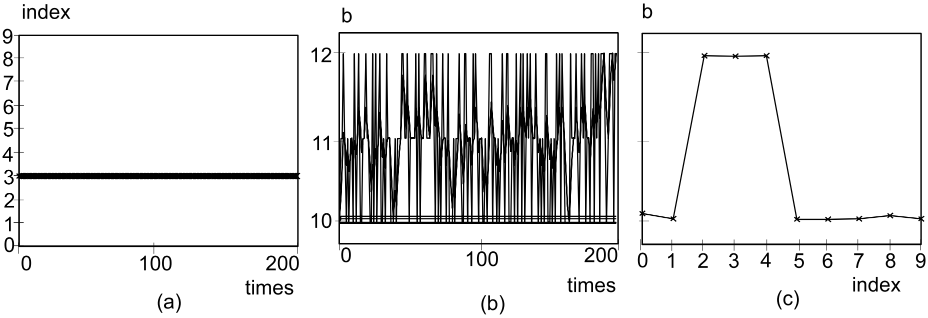


**Fig. 6.** A typical result of the ONLINE simulation of the SSN of the MS model. (a) shows the firing moments $t_{n,i}$ for all $i$ in $t - i$ plane. (b) shows the transition of the dynamic parameter $b_i$ for all $i$ in $i - b$ plane. (c) shows the dynamic parameter $b_i$ at the end of simulation in $i - b$ plane. $[\theta,\ c,\ \alpha,\ L_l,\ L_u,\ \delta w,\ w_c] = [60,\ 40,\ -\pi/60,\ -30,\ 30,\ 5,\ 5]$. Initial dynamic parameter $b_i(0)$ is fixed to $20.0 \cdot 2\pi$ for all $i$. Simulation is done for 200 times and the input intervals are selected randomly from $\{T_A = 2\pi/10, T_B = 2\pi/11, T_C = 2\pi/12\}$.

parisons between the SSN and the BSOM. We simulate one dimensional BSOM to compare the BSOM with the SSN. We select input from $\{10, 11, 12\}$ randomly. We set 10 units in line and initial parameter $m_i(0) = 10$ for all $i$. Fig.5(a) shows time waveforms of the indices of the winner unit. Fig.5(b) shows time waveforms of the reference variable $m_i$ for all neurons. Fig.5(c) shows the reference variable $m_i$ at the end of the simulation. We can see that the reference parameters are clustered in Fig.5(c). From Fig.4 and Fig.5, we can say that the dynamics of the SSN has the similar SOM function like the numerical calculations of the BSOM.

## 4 Comparison with Another Spiking Neuron

We compare the SO model with the Mitubori-Saito (MS) model which is a simple resonate-and-fire model and easily implemented by an electronic circuit. The MS model is the same as the SO model when $\eta > \theta$. Fig.6 shows the result of the SSN simulation using the MS model. We fit the behavior of the MS models to that of the SO models by fitting return maps and bifurcation diagrams. Fig.6(a) shows that, in the SSN of the MS model, the winner never changes and the competitive process doesn't work. By comparing Fig.4 and Fig.6, the typical behavior of the SSN of the MS models is different from that of the SO models. We simulated the SSN of the MS model numerically for many inputs and the results tend to be like Fig.6. We conclude that the SO model is suitable for the SSN compared to the MS model. This is probably because the MS model converges to an equibrium but the SO model doesn't.

## 5 Conclusion

We have presented the subthreshold oscillating (SO) model, the update rule of a dynamic parameter $b$ and the self-organizing network of spiking neurons (SSN). First, we have analyzed the stability of synchronization phenomena of the SO model by return maps. Secondly, we have analyzed the pull-in process of the dynamic parameter $b$. Next, we have shown that the SSN of the SO model and Kohonen's SOM have similar functions and the SSN of the MS model (linear version of the presented SO model) doesn't. Future problems include classification and analysis of bifurcation phenomena and the performance of the network, application to pulse-based signal processing and clarification about the difference between the MS model and SO model on SSN. The authors would like to thank Professor Toshimitsu Ushio of Osaka University for valuable discussions. This work is partially supported by the Center of Excellence for Founding Ambient Information Society Infrastructure, Osaka University, Japan and by KAKENHI (21700253).

## References

1. Hodgkin, A.L., Huxley, F.D.: A quantitative description of membrane current and application to conduction and excitation in nerve. J. Physiol. 117, 500–544 (1954)
2. Izhikevich, E.M.: Resonate-and-fire neuron. Neural Networks 14, 883–894 (2001)
3. Izhikevich, E.M.: Which Model to Use for Cortical Spiking Neurons. IEEE Transactions on Neural Networks 14, 1569–1572 (2003)
4. Izhikevich, E.M.: Dynamical Systems in Neuroscience. MIT Press, Cambridge (2007)

5. Mitsubori, K., Saito, T.: Dependent switched capacitor chaos generator and its synchronization. IEEE Trans. Circuit Syst. I 44(12), 1122–1128 (1997)
6. Miyachi, K., Nakano, H., Saito, T.: Response of a simple dependent switched capacitor circuit to a pulse-train input. IEEE Trans. Circuits Syst. I 50(9), 1180–1187 (2003)
7. Inagaki, T., Saito, T.: Consistency in a chaotic spiking oscillator. IEICE Trans. Fundamentals, E91-A 8, 2040–2043 (2008)
8. Kinoshita, K., Torikai, H.: A spiking neuron model and its pulse-coupled network of a self-organizing map. In: NOLTA 2009 (2009 International Symposium on Nonlinear Theory and its Applications), Paper ID 4040 (2009)
9. Izhikevich, E.M.: Synchronization of MEMS Resonators and Mechanical Neurocomputing. IEEE Transactions on Circuits and Systems I 48, 133–138 (2001)
10. Izhikevich, E.M.: Weakly Pulse-Coupled Oscillators, FM Interactions, Synchronization, and Oscillatory Associative Memory. IEEE Transactions on Neural Networks 10, 508–526 (1999)
11. Nakano, H., Saito, T.: Grouping synchronization in a pulse-coupled network of chaotic spiking oscillators. IEEE Trans. Neural Networks 15(5), 1018–1026 (2004)
12. Kohonen, T.: Self Organizing Maps. Springer, Heidelberg (2001)
13. Hikawa, H.: FPGA implementation of self organizing map with digital phase locked loops. Neural Networks 18, 514–522 (2005)

# A Simple Spiking Neuron with Periodic Input: Basic Bifurcation and Encoding Function

Shimon Teshima and Toshimichi Saito

Hosei University, Koganei, Tokyo, 184-8584 Japan

**Abstract.** This paper studies dynamics of a simple bifurcating neuron to which spike-train input is applied a refractory threshold. The neuron can exhibit rich periodic orbits that is super-stable for initial state and has very fast transient. The periodic orbits are characterized by the rotation number that relates to rate coding of the spike-train. The dynamics can be simplified into the 1-D map of spiking phase. The phase map is piecewise linear and basic bifurcation phenomena can be clarified theoretically.

## 1 Introduction

The bifurcating neuron (BN, [1]-[5]) is a switched dynamical system based on artificial spiking neuron models. The BN has a periodic base signal and the state variable repeats integrate-and-fire behavior between the threshold and base. Depending on the shape of the base, the BN can exhibit a variety of periodic/chaotic spike-trains and related rich bifurcation phenomena. Using plural BNs, pulse-coupled networks (PCN) can be constructed. The PCN can exhibit rich synchronous/asynchronous phenomena.

Motivations for studying the BN and PCN are many, include the following three points: (1) analysis of spike-trains, synchronization and bifurcation phenomena is important as basic nonlinear problems [8]-[10]; (2) analysis of spike-based encoding function is a key to understand information processing function in the brain [6] [7]; (3) spike-trains are low power and their rich dynamics can be basic to develop novel engineering applications such as spike-based communications [3], spike-based signal processing [11]-[14] and associative memories [4]. However, analysis of the BN has not yet completed because of complex nonlinearity.

This paper studies dynamics of a simple BN with spike-train input. First, as a preparation, we introduce a BN with sawtooth-type base signal. The dynamics can be simplified into the 1-D map of spiking phase. The phase map is piecewise linear and basic bifurcation phenomena can be clarified precisely. We then apply periodic spike-train input to the BN via a refractory threshold. For simplicity we consider the case of the dc base signal. The refractory threshold can cause rich periodic orbits that is super-stable for initial state and has very fast transient. In order to characterize the super-stable periodic orbits (SSPO), we introduce

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 376–383, 2009.

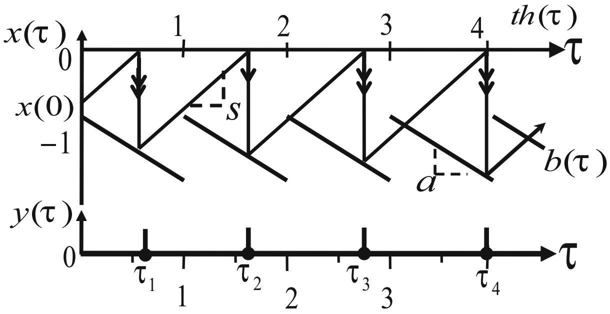


**Fig. 1.** Dynamics of BN

the rotation number that relates to rate coding of the spike-train. The phase map is equivalent to the circle map with flat segment and basic bifurcation can be analyzed theoretically. Especially, it is clarified that the rotation number exhibits one-direction devil's stair for a parameter. These novel results are basic to clarify spike-based encoding function of wider class of BNs and PCNs; and can be a trigger to develop novel bifurcation theory of SSPOs. It should be noted that there exist few works that bridge between nonlinear neuron dynamics and analog signal processing.

## 2 Bifurcating Neurons

Fig.1 shows dynamics of the BN. The state variable $x$ rises to the threshold $th$. When it reaches $th$, the threshold it jumps to $b(\tau)$ and BN outputs a spike $y = 1$. Repeating this behavior the BN outputs spike-train $y(\tau)$. The dynamics is described by Eq.(1).

$$\begin{cases} \frac{dx}{d\tau} = s & \text{and} \quad y(\tau) = 0 \quad \text{for} \quad x(\tau) < th = 0 \\ x(\tau_+) = b(\tau_+) & \text{and} \quad y(\tau_+) = 1 \text{ if} \quad x(\tau) = th \end{cases} \tag{1}$$

where $\tau$ denotes the dimensionless time and $th = 0$ is the normalized threshold. Adjusting the shape of $b(\tau)$, the BN can output a variety of spike-trains and exhibits rich bifurcation phenomena. For simplicity, we use the sawtooth base signal with period 1:

$$\begin{aligned} &b(\tau + 1) = b(\tau), \\ &b(\tau) = a(\tau - 0.5) - 1 \;\text{ for }\; 0 \le \tau < 1. \end{aligned} \tag{2}$$

This BN is characterized by two parameters $s$ and $a$ that control rising slope of $x$ and amplitude of $b(\tau)$, respectively. For simplicity we focus on the following parameter subspace.

$$P_s \equiv \{(a, s)|0 < s, -2 < a < 2\}$$

In order to consider the behavior of the BN, we derive a spike-position map. Let $\tau(n)$ denote the $n$-th spike-position where $n$ is a positive integer. Since $\tau(n)$ determines $\tau(n+1)$, we can define a spike-position map:

$$\tau(n+1) = f(\tau(n)) \equiv \tau(n) - \frac{1}{s}b(\tau(n)) \tag{3}$$

Since the base signal is period 1, we can introduce the spike-phase $\theta(n) \equiv \tau(n) \mod 1$. Substituting Eq. (2) into (3) and using the spike phase, we can describe the phase map as the following:

$$\begin{aligned} &\theta(n+1) = F(\theta(n)) \equiv f(\theta(n)) \mod 1 \\ &= (1 - \tfrac{a}{s})(\theta(n) - \tfrac{1}{2}) + \tfrac{1}{s} + \tfrac{1}{2} \mod 1 \end{aligned} \tag{4}$$

As shown in Fig. 2, we define some parameter sets:

$$\begin{aligned} &P_{spo} \equiv \{(a,s)|0 < a < 2s\} \cap P_s \\ &P_{c1} \equiv \{(a,s)|2s < a\} \cap P_s \\ &P_{c2} \equiv \{(a,s)|a < 0\} \cap P_s \\ &P_{rot} \equiv \{(a,s)|a = 0\} \cap P_s \\ &P_{cut} \equiv \{(a,s)|a = -s\} \cap P_s \end{aligned}$$

In $P_{spo}$, the BN outputs periodic spike-train corresponding to the phase map as shown in Fig. 2 (b). In $P_{c1} \cup P_{c2}$, the BN outputs chaotic spike-train corresponding to the phase map as shown in Fig. 2 (a) and (d). In $P_{rotation}$, $F$ is onto and is equivalent to the rotation of the unit circle [8] as shown in Fig. 2 (c): $F(\theta) = \theta + 1/s \mod 1$. In this case $F$ exhibits a variety of periodic or quasi-periodic orbit depending on parameter value. $1/s$ that corresponds to the rotation number. The rotation map relates deeply to the sigma-delta modulator, a simple A/D converter with rate coding. In $P_{c2}$, the map $F$ is equivalent

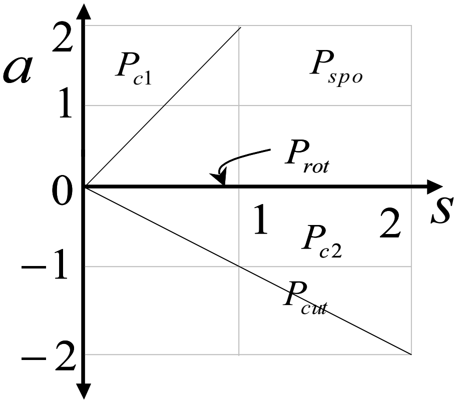


**Fig. 2.** Some parameter sets

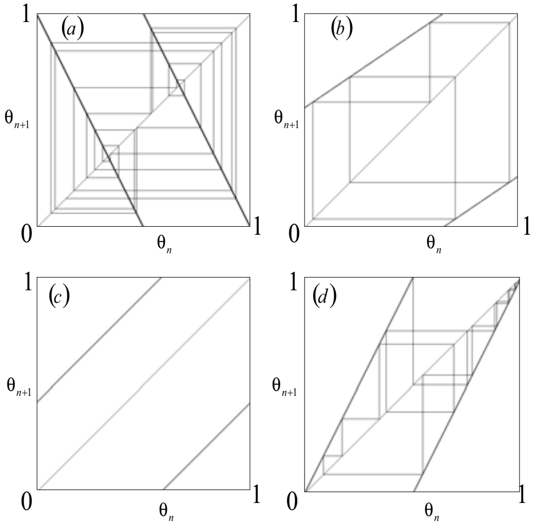


**Fig. 3.** Typical return maps. (a) $a = 1.2, s = 0.4$ (b) $a = 0.4, s = 2.4$ (c) $a = 0, s = 2.4$ (d) $a = 2/3, s = 2/3$.

to the cut map that can be regarded as a binary encoder. Fig.3 shows typical phase maps. The phase map is piecewise linear and we can analyze the dynamics precisely.

## 3 BN with Periodic Spike-Train Input

Let us consider the case of $a = 0$ with periodic spike-train input. This is a basic system to consider response characteristics of the BN from spike-train inputs. As shown in Fig. 5. the base signal is constant, $b(\tau) = -1$, and $x$ rises with slope $s$ to the normalized threshold $th$=0. The periodic spike-train with period 1 is applied to the BN. The state variable $x$ is reset to the base either (1) $x$ reaches the threshold $th$=0, or (2) a spike arrives during $x$ exceeds the refractory threshold $-q$. The dynamics is described by

$$\begin{cases} x(\tau_+) = -1 & \text{if} \quad x(\tau) = 0 \\ x(\tau_+) = -1 & \text{if} \quad x(\tau) \geq -q \text{ and} \quad \tau = n \\ \dfrac{dx}{d\tau} = s, \quad \text{otherwise} & \end{cases} \tag{5}$$

where $-1 < q < 0$. The refractory threshold $-q$ controls effects of the input on the resetting: as $-q$ approaches to 1, the effects of the input decreases. Spike-position map and spike-phase map are given by the following equation.

$$\tau_{n+1} = f(\tau) \\ = \begin{cases} m & \text{for} \quad m - \dfrac{1}{s} \leq \tau(n) < \dfrac{q}{s} - \dfrac{1}{s} + m \\ \tau(n) + \dfrac{1}{s} + 1, & \text{otherwise} \end{cases} \tag{6}$$

$$\theta(n+1) = F(\theta(n)) = f(\theta(n)) \bmod 1 \\ = \begin{cases} \theta(n) + \frac{1}{s}, & \text{for} \quad \theta(n) \in I_1 \equiv \{\theta(n) | 0 \leq \theta(n) < 1 - \frac{1}{s}\} \\ 0, & \text{for} \quad \theta(n) \in I_2 \equiv \{\theta(n) | 1 - \frac{1}{s} \leq \theta < 1 - \frac{1}{s} + \frac{q}{s}\} \\ \theta(n) + \frac{1}{s} - 1, & \text{for} \quad \theta(n) \in I_3 \equiv \{\theta(n) | 1 - \frac{1}{s} + \frac{q}{s} \leq \theta(n) < 1\} \end{cases} \tag{7}$$

where $m$ denote positive integers. Fig.5 shows typical examples of $F$. $F$ is a rotation map with the trapping window $I_2$ that has rich periodic orbits. Hereafter we refer to $F$ as rotation with window width $\frac{q}{s}$. Note that an orbit must enter into the window $I_2$ if $1/s$ is irrational. Since slope of $F$ is zero on $I_s$, the periodic orbit is superstable for initial state and has very fast transient. In order to characterize the SSPO, we introduce an encoding:

$$\omega(\theta(n)) = \begin{cases} 0 & \text{if} \quad \in I_1 \\ 1 & \text{if} \quad \in I_2 \bigcup I_3 \end{cases} \tag{8}$$

A periodic orbit with period $M$ is characterized by a rotation number:

$$\rho = \frac{1}{M} \sum_{n=1}^{M} \omega(\theta(n)) \tag{9}$$

If $q = 0$ and an orbit is quasi-periodic then the rotation number is given by the limit for $M \to \infty$. Fig.6 illustrates a rough bifurcation diagram. The BN exhibits SSPO and the period increases as the tone becomes dark. In order to clarify existence region of SSPOs we define parameter sub-sets:

$$P_{\frac{N}{M}} \equiv \{(a, s) | \frac{N}{M} \leq \frac{1}{s} < \frac{N}{M - q}\} \tag{10}$$

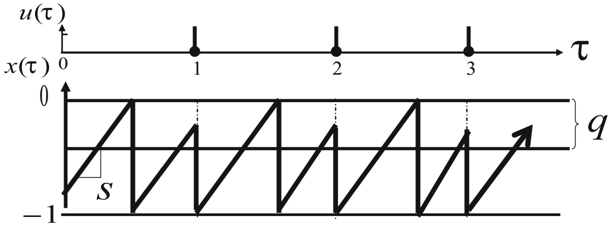


**Fig. 4.** Dynamics of BN with periodic pulse-train input

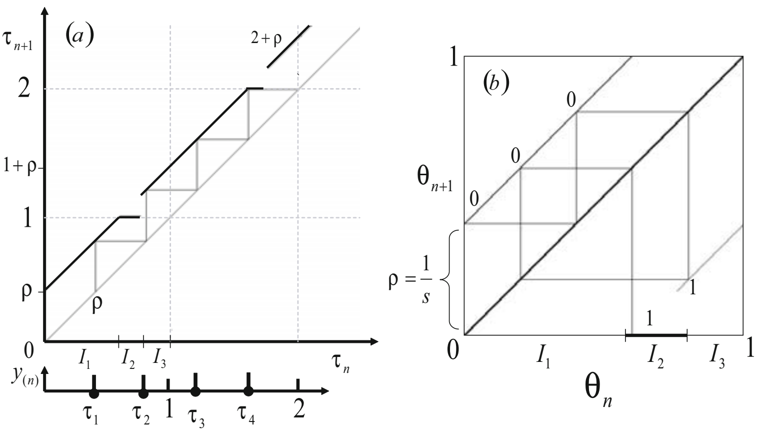


**Fig. 5.** Key mappings. (a) spike-position map, (b) spike-phase map.

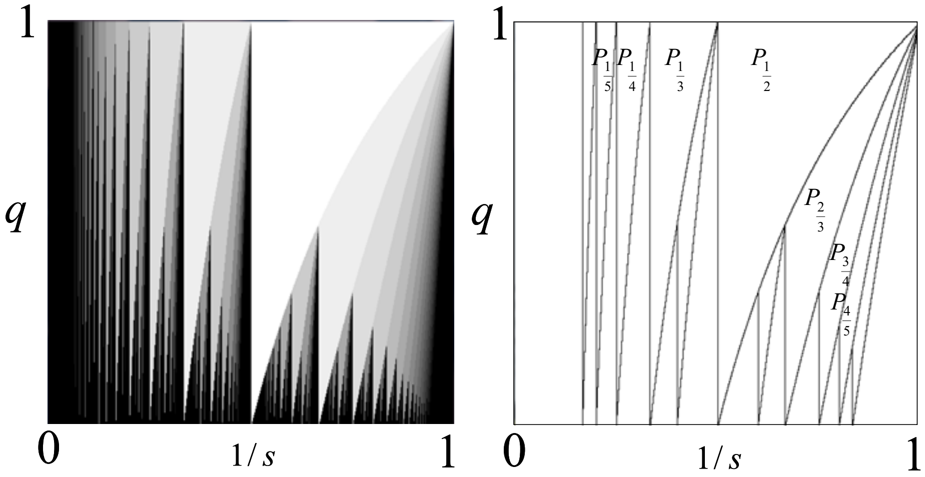


**Fig. 6.** (a)Bifurcation diagram. Tone becomes dark as period of SSPO increases. (b) parameter region $P_{\frac{N}{M}}$.

*Theorem*: The SSPO with rotation number $N/M$ exists if $(a, s) \in P_{\frac{N}{M}}$ and $(a, s) \notin P_{\frac{N'}{M'}}$ for integers $M'$ and $N'$ such that $M' < M$ and $N' < M'$.

This is a theoretical background for Fig.6. Fig. 7 shows encoding characteristics where we regards $1/s$ and $\rho$ as input and output, respectively. If $q = 0$, the map $F$ is rotation and the characteristics Fig.7(a) for a finite value of $l$ corresponds to basic analog-to-digital conversion by rate coding. For $q > 0$, the map $F$ exhibits SSPOs (with finite period) for almost all parameter values. Using encoding Eq. (8), we can transform the SSPO into binary sequence

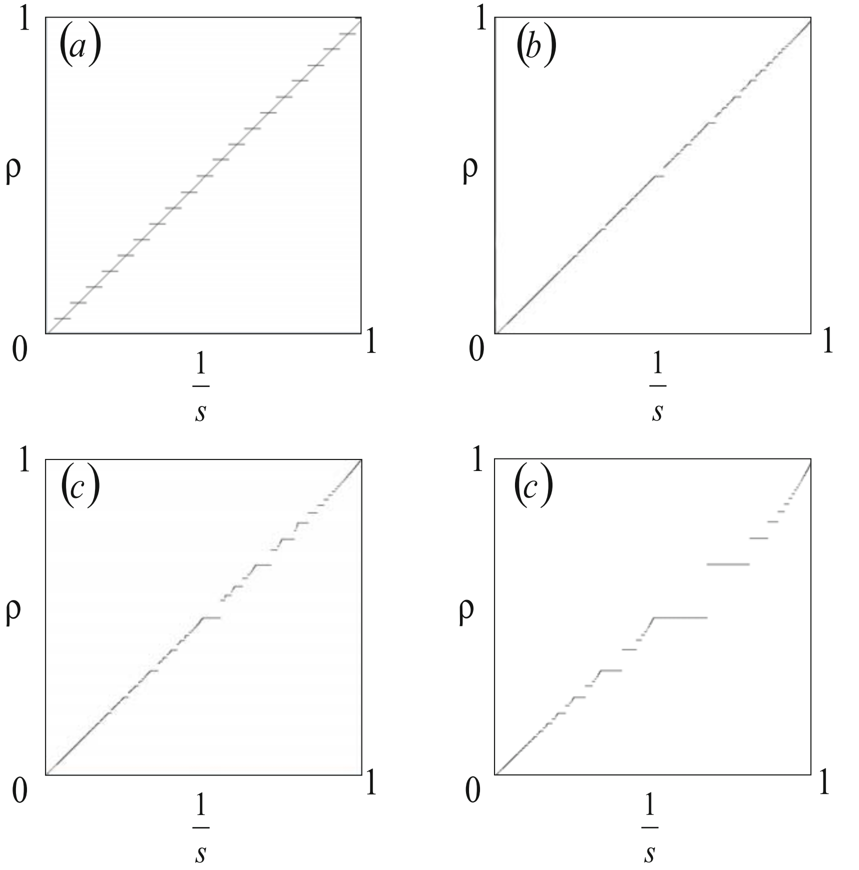


**Fig. 7.** Encoding characteristics. (a) $q = 0, l = 20$ (b) $q = 0.1, l = \infty$ (c) $q = 0.2, l = \infty$ (d) $q = 0.5, l = \infty$.

with finite code length. In this case, the rotation number corresponds to approximation of the analog input $1/s$. As $q$ approaches zero, variety of SSPOs and encoding resolution increase. The charactersitics seem to be one-direction devils's stair case. This is a basic result to consider encoding function of the BNs.

## 4 Conclusions

We have analyzed basic bifurcation phenomena of simple BM with spike-train input. The refractory threshold causes rich SSPOs. The SSPOs are characterized by the rotation number of rate encoding. Using the phase map, bifurcation can be clarified precisely.

Future problems include bifurcation analysis for wider class of base signals, analysis of spike-based encoding function, and design of simple hardware [3] [15] for engineering applications.

## References

1. Perez, R., Glass, L.: Bistability, period doubling bifurcations and chaos in a periodically forced oscillator. Phys. Lett. 90A(9), 441–443 (1982)
2. Allen, T.: Complicated, but rational, phase-loching responses of a simple time-base oscillatos. IEEE Trans. Circuits Syst. 30(9), 627–632 (1983)
3. Torikai, H., Saito, T., Schwarz, W.: Synchronization via multiplex pulse-train. IEEE Trans. Circuits Syst. I 46(9), 1072–1085 (1999)
4. Lee, G., Farhat, N.H.: The bifurcating neuron network 1. Neural networks 14, 115–131 (2001)
5. Hernandez, E.D.M., Lee, G., Farhat, N.H.: Analog Realization of Arbitrary One-Dimensional Maps. IEEE Trans. Circuits Syst. I 50(12), 1538–1547 (2003)
6. Izhikevich, E.M.: Simple Model of Spiking Neurons. IEEE Trans. Neural Networks 14(6), 1569–1572 (2003)
7. Toyoizumi, T., Aihara, K., Amari, S.: Fisher information for spike-based population coding. Phys. Rev. Lett. 97, 098102 (2006)
8. Kon'no, Y., Saito, T., Torikai, H.: Rich dynamics of pulse-coupled spiking neurons with a triangular base signal. Neural Networks 18, 523–531 (2005)
9. Shimazaki, M., Torikai, H., Saito, T.: Synchronization and window map from pulse-coupled relaxation oscillators. IEICE Trans. Fundamentals, E87-A 9, 2426–2431 (2004)
10. Patel, A., Kosko, B.: Stochastic Resonance in Continuous and Spiking Neuron Models With Levy Noise. IEEE Trans. Neural Networks 19(12), 1993–2008 (2008)
11. Chen, K., Wang, D.L.: A dynamically coupled neural oscillator network for image segmentation. Neural Networks 15, 423–439 (2002)
12. Hamanaka, H., Torikai, H., Saito, T.: Quantized spiking neuron with A/D conversion functions. IEEE Trans. Circuits Syst. II 53(10), 1049–1053 (2006)
13. Hopfield, J.J., Herz, A.V.M.: Rapid local synchronization of action potentials: Toward computation with coupled integrate-and-fire neurons. Proc. Natl. Acad. Sci. USA 92, 6655–6662 (1995)
14. Rowcliffe, P., Feng, J., Buxton, H.: Spiking Perceptrons. IEEE Trans. Neural Networks 17(3), 803–807 (2006)
15. Vogelstein, R.J., Mallik, U., Vogelstein, J.T., Cauwenberghs, G.: Dynamically Reconfigurable Silicon Array of Spiking Neurons With Conductance-Based Synapses. IEEE Trans. Neural Networks 18(1), 253–265 (2007)

# Exploiting Temporal Noises and Device Fluctuations in Enhancing Fidelity of Pulse-Density Modulator Consisting of Single-Electron Neural Circuits

Andrew Kilinga Kikombo*, Tetsuya Asai, and Yoshihito Amemiya

Graduate School of Information Science and Technology, Hokkaido University,
060-0814, Sapporo, Japan
{kikombo,asai,amemiya}@sapiens-ei.eng.hokudai.ac.jp

**Abstract.** This paper discusses the implications of noises in a pulse-density modulation single-electron circuit based on Vestibulo-ocular Reflex model. The proposed circuit consists of an ensemble of single-electron integrate-and-fire neurons that encode the input voltage into pulses whose temporal density is proportional to the amplitude of the input. We confirmed that static noises (heterogeneity in circuit parameters) and dynamic noises (random firing) introduced into the network indeed played an important role in improving the fidelity with which the neurons could encode signals with input frequencies higher than the intrinsic response frequencies of single neurons or a network of neurons without noises. Through Monte-Carlo based computer simulations, we demonstrated that noises could enhance the fidelity with which the network could correctly encode signals with high input frequencies: a noisy network could operate over a wider input range that a single neuron or a network of homogeneous neurons.

**Keywords:** neuromorphic LSIs; neural networks; single-electron circuits; pulse-density modulation.

## 1 Introduction

As physical features of electronic devices approach the deep sub-micron (nano) scales, variance in physical parameters of fabricated devices (static noises) and sensitivity to external noises (dynamic noises) become more pronounced, posing a challenging task to the circuit designer. Most of the research toward solving these problems has been focused on reducing the impact of static noises through improved fabrication techniques, improving shielding technologies to protect devices from radiation and extrenal noises, or architectural level approaches where additional circuitry is introduced into the system to increase the signal to noise ratio. The above-mentioned approaches might not provide the once-and-future solution, especially for the constantly shrinking device sizes. A novel approach to

* Corresponding author.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 384–391, 2009.

solving this problem would be to effectively use both static and dynamic noises to improve circuit performance.

If we look at how signal processing is carried out in neuronal systems, we find that individual neurons have high heterogeneity in intrinsic response properties; they have diverse variances in firing rates, and some of the neurons are even defective. However, in spite of these setbacks neuronal systems accurately encode signals as they are relayed from sensory organs to the central nervous system, or to other organs. A number of reports suggest that neurons in fact employ heterogeneity to effectively encode signals ([1] - [3]). Hospedales et al. ([1]) demonstrated that neurons in the medial vestibular nucleus (MVN) can encode high frequency signals with a high temporal precision as a result of their heterogeneity. This paper introduces a neuromorphic circuit that effectively utilizes both static and dynamic noises to improve the temporal fidelity of signal transmission in a pulse density modulation circuit based on Vestibulo-ocular Reflex (VOR) model.

The paper is organised as follows. A short review of pulse-density modulation in integrate-and-fire neurons is presented. This is folowed by the noisy network model and its circuit implementation with single-electron devices. Thirdly, the model and circuit configuration are explained. Finally the validity of the model is verified with Monte-Carlo based simulations.

## 2 Pulse-Density Modulation in Integrate-and-Fire Neurons

An integrate-and-fire neuron (IFN) aggregates inputs from other neurons connected through synapses. The aggregated charge raises the membrane potential until it reaches a threshold, where the neuron fires generating a spike. This spike

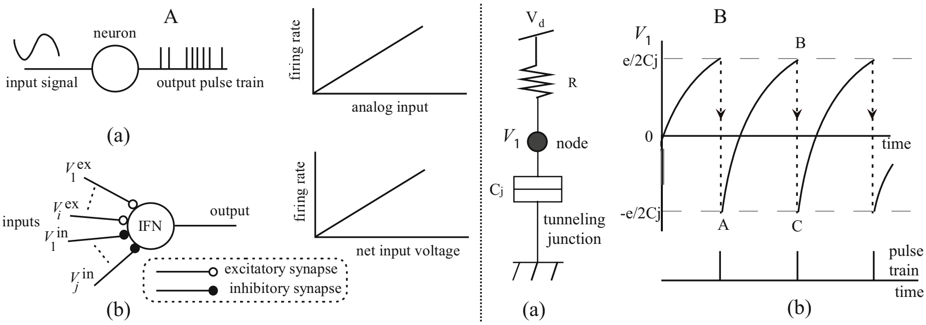


**Fig. 1.** A:(a) Pulse density modulation in integrate-and-fire neurons: analog input is converted into a pulse train (b) Fundamental structure and operation of integrate-and-fire neurons (IFNs). The IFN receives input voltages through excitatory and inhibitory synapses, and produces a pulse train whose pulse density (firing rate) is proportional to the net input voltage. B: Single-electron tunneling (SET) oscillator: (a) circuit structure and (b) waveform showing oscillation.

corresponds to a binary output g1h. After the firing event, the membrane potential is reset to a low value, and it increases again as the neuron accepts inputs from neighboring neurons (or input signals) to repeat the same cycle; producing a stream of goneh and gzeroh pulse trains. The spike interval (density of spikes per unit time) is proportional to the analog input voltage i.e. the level of analog input is coded into pulse density. Thus a neuron can be considered as a 1-bit A-D converter operating in the temporal domain. Fig. 1A:(a) shows a schematic representation of analog-to-digital conversion in IFNs. The output pulse density is proportional to the amplitude of the input signal. Fig. 1A:(b) shows the fundamental operation of an IFN. The open circles (○) and shaded circles (●) represent excitatory and inhibitory synapses, respectively. The IFN receives input signals (voltages) through the excitatory synapses (to raise its membrane potential) and inhibitory synapses (which decrease the membrane potential) from adjacent neurons, to produce a spike if the postsynaptic potential ($\sum V_i^{\mathrm{ex}} - \sum V_j^{\mathrm{in}}$) exceeds the threshold voltage. After the IFN fires, its membrane voltage is reset to a low value, and the integration action resumes.

## 3 Single-Electron Integrate and Fire Neuron

The operation of an integrate-and-fire neuron (IFN) is modelled with a single-electron oscillator [4] - [5]. A single-electron oscillator (Fig. 1B:(a)) consists of a tunneling junction (capacitance = $C_{\mathrm{j}}$) and a high resistance $R$ connected in series at the nanodot (●) and biased with a positive or a negative voltage $V_{\mathrm{d}}$. It produces self-induced relaxation oscillations if the input voltage is higher than the tunneling threshold ($V_{\mathrm{d}} > e/(2C_{\mathrm{j}})$) where $e$ is the elementary charge and $k_B$ is the Boltzmann constant. The naodot voltage $V_1$ increases as the capacitance $C_{\mathrm{j}}$ is charged through the series resistance (curve AB), until it reaches the tunneling threshold $e/(2C_{\mathrm{j}})$, at which an electron tunnels from the ground to the nanodot across the tunneling junction, resetting the nanodot voltage to $-e/(2C_{\mathrm{j}})$. This abrupt change in nanodot potential (from B to C) can be referred to as a firing event. The nanodot is recharged to repeat the same cycles. Therefore, a single-electron oscillator could be viewed as an integrate and fire neuron, which aggregates inputs (or inputs from from neighboring neurons) producing a pulse when its nanodot voltage reaches the threshold voltage (Fig. 1B:(b)). By feeding a sinusoidal input to a single-electron oscillator, one can adjust the probability of electron tunneling in the circuit: the tunneling rate increases as the input voltage rises above the threshold and gradually decreases to zero as the input approaches and falls below the threshold value. In other words, a single-electron oscillator converts an analog input into digital pulses. A single-electron oscillator can thus be viewed as a pulse-density modulator (PDM), that produces a spike train (or produces zero) if the input signal exceeds (or falls below) the threshold value.

## 4 Model and Circuit Structure

The single-electron integrate-and-fire neuron explained in the preceding section is used to study the implications of noises enhancing fidelity of signal transmission in a neuronal single-electron circuit. The circuit is based on a model of the vestibulo-ocular reflex (VOR) proposed by Hospedales et al. ([1]). In their work, they reported that noises and heterogeneity in the intrinsic response properties of neurons account for the high-fidelity in VOR functionality.

Fig. 2(a) shows the part of the model, which converts head movements into neural spikes in the VOR, consisting of $n$ neurons. The structural heterogeneity in the synaptic couplings (membrane time constants) of individual neurons is represented by $\xi_i$. We refer to this heterogeneity as static noises. The neurons receive a common analog input and produce spikes whose temporal density corresponds to the amplitude of the input signal. The output terminal receives pulses from all the neurons in the network to produce a spike train. The noises introduced into the network lead to random and independent firing events in the neurons, reducing the probability of synchrony in the network. In addition, the variations in parameters increases the randomness with which the network neurons fire, increasing the probability of a ready-to-fire neuron at any given time, which consequently enhances the precision with which the neurons in the network can encode signals with input frequencies higher those of individual neurons.

The network is implemented with single-electron IFNs (oscillators) as shown in Fig. 2(b). The heterogeneity in the model was introduced in the circuit as variations in the series resistance $R$. Note that $R$ is a critical parameter in setting the intrinsic response frequency of each neuron. Therefore, by tuning the values of $R$, we could simulate the heterogeneity of membrane time constants of actual neurons.

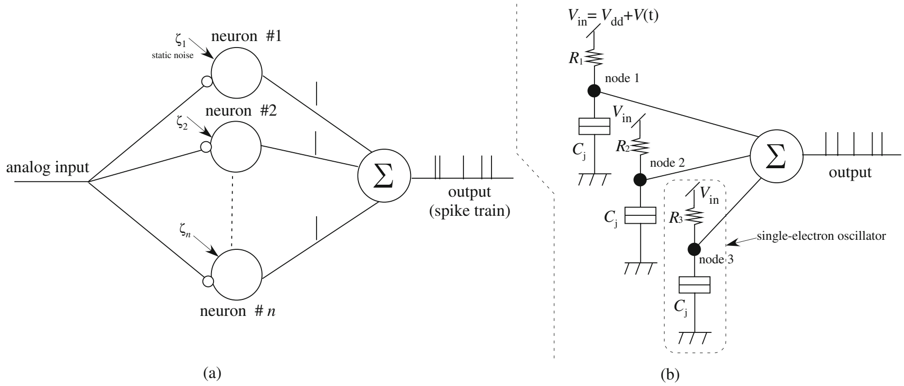


**Fig. 2.** (a) Neural network model of signal encoding in the VOR consisting of $n$ neurons, (b) Implementation with single-electron oscillators

## 5 Simulation Results

In the simulations, the single-electron neurons were fed with a common input voltage $V_{\rm in} = V_{\rm dd} + V({\rm t})$, where $V_{\rm dd}$ (bias voltage) was set to 7.8 mV to achieve a monostable operation in the absence of input signals, $V({\rm t})$ is a pulsed input voltage with an amplitude of 0.8 mV. The capacitance of the tunneling junctions $C_{\rm j}$ was set to 10 aF. The simulation time was set to 800 ns, while the operation temperature $(T)$ was set to 0.5 K for simulation results shown in Figs. 3, 4 and 5(b) and (c).

Fig. 3 shows the transient response of a unit single-electron neuron. Fig. 3(a) and (c) show the input signals with a frequency of 600 MHz and 250 MHz, respectively. Fig. 3(b) shows the neuron response to input "(a)", while "(d)" shows the neuron respone to input "(c)". The series resistance was set to 100 MΩ. Fig. 3(d) shows successful encoding of the input signal (the neuron fires once for each pulse in the input signal[1]) whose frequency is within the intrinsic firing rate of a single neuron. In Fig. 3(b), the neuron could only trasmit some of the input pulses, leading to a lower firing rate as compared to the input rate. In other words, the neuron in (b) could only transmit some of the input pulses toward the output. This degrades the fidelity of signal transmision along the neural network.

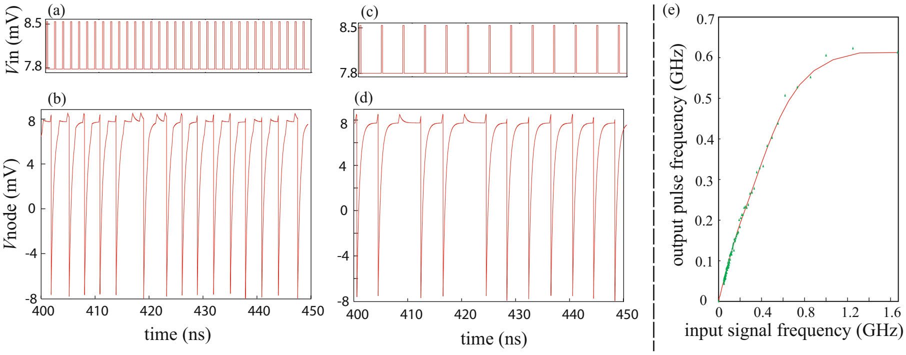


**Fig. 3.** Transient response of a single neuron. (a) and (c) show input signals with input frequencies of 600 MHz and 250 MHz, respectively. (b) and (d) show the output characteristics of neurons fed with input signals of 600 MHz and 250 MHz, respectively. (e) Output firing rate of a single neuron plotted against the input pulse frequency.

[1] Tunneling (firing) in single-electron devices involves a probabilistic time lag or waiting time between when the node voltage exceeds the threshold voltage and when an electron can actually tunnel from the ground to the node, sending a spike toward the output terminal. Due to the effect of the time lag, a neuron might fail to fire even after achieving the tunneling conditions as seen in Fig. 3(d). As a result, the average firing rate would be somewhat lower than the input pulse rate.

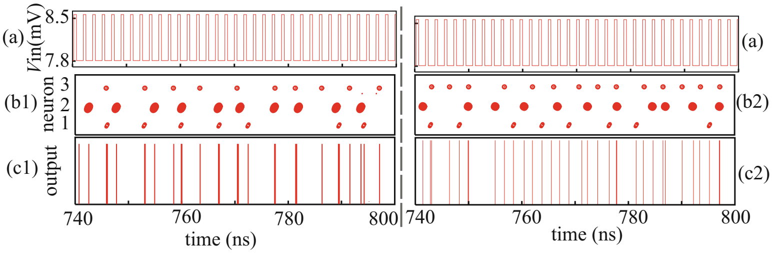


**Fig. 4.** Transient responses of both homogeneous and heterogeneous networks. (a) shows the input signal. (b1) shows the firing events of each neuron, while (c1) whows the summed pulse output for the three neurons in the homogeneous network. (b2) shows the firing events, and (c2) shows the summed pulse output of the heterogeneous network.

Fig. 3(e) shows the response of a single neuron over a wide range of input frequencies. The horizontal axis shows the input frequency, while the vertical axis shows the average firing rate of the neuron. The neuron response was linear for input signals with a frequency of upto 0.5 GHz. Beyond this range, the output was highly distorted. This shows that a single neuron can successfully encode (respond to) signals with a maximum input frequency of 500 MHz.

The response of a population of neurons to various input frequencies was investigated with two sets of neuron ensembles: homogeneous and heterogeneous networks. In the homogeneous ensemble, the series resistances $R_1$, $R_2$, and $R_3$ were set to the same value, whereas in the second set, heterogeneity (static noises) was introduced by varying the values of series resistances in the three neurons. The results are shown in Figs. 4 and Fig. 5.

Fig. 4(a) shows the input signal with a frequency of 600 MHz. Figs. 4(b1) and (c-1) show the response of the homogeneous network, where the series resistances $R_1$, $R_2$ and $R_3$ were set to 100 MΩ. Fig. 4(b1) shows the firing events of individual neurons in the network. Fig. 4(c1) shows the summed spike output (spike train) at the output terminal. We could confirm that the neurons in the homogeneous network tend to synchronize, firing at almost the same timing.

Figs. 4 (b2) and (c2) show the response of neurons in the heterogeneous network, where the series resistances were set to 110 MΩ for neuron 1, 100 MΩ for neuron 2 and 90 MΩ for neuron 3. The firing events in the heterogeneous network are more or less random as shown in Fig. 4(b2). The probability of having a neuron with a potential near the threshold value, at any given moment, is higher than in the case of a homogeneous network. Thus the network can respond to any incoming pulses at a higer probability. This results in an improved encoding of the input as illustrated by the spike train shown in Fig. 4(c2). In other words, since the neurons fired irregularly, they could transmit the input pulses with a higher temporal precision as opposed to the homogeneous network. This is elaborated in more detail in Fig. 5 (curves (b) and (c)), where the transmission of signal over a wide range of frequencies is demonstrated. The horizontal

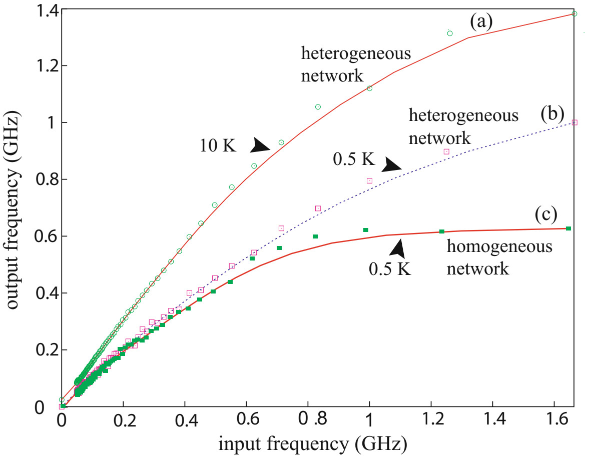


**Fig. 5.** Output firing rate of an ensemble of neurons plotted against the input pulse frequency. (a) and (b) show response characteristics of a heterogeneous network simulated at a temperature of 10 K and 0.5 K, respectively. (c) shows response characteristics of a homogeneous network simulated at 0.5 K.

axis represents the frequency of input signals, while the vertical axis shows the average firing rate (output frequency) for both neuron sets. In the case of the homogenous network, since the neurons tend to synchronize with time, their encoding frequency is the same as that of individual neurons. Contrary, neurons in the heterogeneous network could correctly encode signals with input frequencies upto 1 GHz, twice that of the homogeneous network. This demonstrates that heterogeneity in the circuit parameters (presence of static noises) plays an important role in improving the fidelity with which neurons can encode signals with input frequencies far beyond the encoding capacity of individual neurons.

## 6 Effect of Dynamic Noises

Hospedales et al. ([1]) investigated the importance of random noises in improving the fidelity of signal transmission in the VOR response. They concluded that besides neuronal heterogeneity, externally induced noises also play an important role in improving the network performance. These external noises could be as a result of spontaneous increases or decreases of membrane potential due to firing events in other neurons in the network. These changes are random and are often referred to as dynamic noises. In our circuit, we studied the effect of dynamic noises by considering thermally induced tunneling events in the network. Curves(a) and (b) in Fig. 5 show the response characteristics of a network simulated at 10K, and 0.5K, respectively. As the temperature increases, thermally induced tunneling events in single-electron neurons increase, resulting in an increase in the average firing rate in the network. This is illustrated by the increased firing rate at a temperature of 10

K. Although this work suggests that dynamic noises don't play a critical role in increasing the maximum response frequency of the network, they however, increase the fidelity with which the network can sample input signals within the maximum input signal frequency range determined by heterogeneity in the network elements. This is evident at higher input frequencies, where the ratio of the output pulse rate to the input pulse rate starts to roll-off rapidly. The roll off is compensated for by the dynamic noises, which reduces the effect of waiting time in electron tunneling.

## 7 Conclusion

In this study, we proposed and investigated the implication of heterogeneity in transmission of high frequency signals in a neural network. Through Monte-Carlo based computer simulations, we confirmed that heterogeneity in device parameters indeed improved the temporal precision with which the network could transmit signals with high input frequencies within the network. A heterogeneous network could correctly encode signals of upto 1 GHz, as compared to 500 MHz in single neurons (or a network of homogenous neurons). Another important factor to consider in improving the fidelity of this circuit would be the effect of external and internal (dynamic) noises. In single-electronic devices, such noises include thermally induced random firing events or the effect of environmental noises. As we have shown, as the temperature increases, the dynamic noises also increase compensate for the roll-off in response of the network, especially at high frequencies. Although a comprehensive investigation on the implications of dynamic noises to signal transmission is required, the preliminary results presented in this paper show that in addition to heterogeneity in neuron properties, externally introduced noises could assist in further improving the fidelity of signal encoding in single-electron circuits. We should however, note that at higher temperatures, beyond the results presented here, random tunneling as a result of dynamic noises would increase rapidly leading to degradation of signal transmission. Therefore, the value of dynamic noises to be introduced to the network to achieve the best performance needs to be optimized.

## References

1. Hospedales, T.M., Rossum, M.C.W., Graham, B.P., Dutia, M.B.: Implications of Noise and Neural Heterogeneity for Vestibulo-Ocular Reflex Fidelity. Neural Computation 20, 756 (2008)
2. Mar, D.J., Chow, C.C., Gerstner, W., Adams, R.W., Collins, J.J.: Noise shaping in populations of coupled model neurons. PNAS 96, 10450 (1999)
3. Oya, T., Asai, T., Amemiya, Y.: Stochastic resonance in an ensemble of single-electron neuromorphic devices and its application to competitive neural networks. Chaos, Solitons & Fractals 32, 855 (2007)
4. Grabert, H., Devoret, M.H.: Single Charge Tunneling—Coulomb Blockade Phenomena in Nanostructures. Plenum Press, New York (1992)
5. Oya, T., Asai, T., Fukui, T., Amemiya, Y.: Reaction-Diffusion Systems Consisting of Single-Electron Oscillators. Int. J. Unconventional Computing 1, 177 (2005)

# Bifurcation Analysis of a Resonate-and-Fire-Type Digital Spiking Neuron

Tetsuya Hishiki and Hiroyuki Torikai

Graduate School of Engineering Science, Osaka University,
1-3 Machikaneyama, Toyonaka, 560-8531 Japan
hishiki@hopf.sys.es.osaka-u.ac.jp, torikai@sys.es.osaka-u.ac.jp

**Abstract.** In this paper we present a generalized model of a resonate-and-fire-type digital spiking neuron. Depending on a stimulation input, the neuron generates various firing spike-trains. Using a hybrid return map, we can analyze typical bifurcation phenomena of the neuron. The analysis clarifies that the generalized model can mimic many responses of biological spiking neurons.

## 1 Introduction

Various spiking neuron models have been proposed in order to develop artificial pulsed neural networks and to explore biological neuron dynamics [1]-[10]. As shown in Table 1, the integrate-and-fire model has 1-dimensional continuous state dynamics with reset [1],[2],[4]. The resonate-and-fire model has 2-dimensional continuous state dynamics with reset [1],[2],[5]. They are analog models, so dynamic parameter adjustments of the neuron models that are implemented in electronic circuits are difficult.

On the other hand, we have been proposed the integrate-and-fire-type digital spiking neurons which have digital state dynamics [8],[9]. The digital spiking neurons consists of shift registers. These models can change their parameter values (wirings among the registers) dynamically on a reconfigurable device such as FPGA. In this paper, first we present a resonate-and-fire-type digital spiking neuron which is a generalized model in [10]. Next we derive a hybrid return map which describes the dynamics of the neuron, and using the map we analyze the dynamics without any approximations.

Significances and novelties of this paper include the following points. (a) The analysis clarifies that the generalized model can exhibits richer dynamics than the old models.

**Table 1.** Difference between the proposed neuron and other models. IF: integrate-and-fire, RF: resonate-and-fire, NA: non-autonomous.

| Models | Dynamics | Behavior | Dynamic parameter adjustment |
|---|---|---|---|
| Integrate-and-fire [1],[2],[4] | 1-D ODE + reset | IF, NA IF | × |
| Resonate-and-fire [1],[2],[5] | 2-D Linear-ODE + reset | IF, RF | × |
| Izhikevich [1],[2] | 2-D Nonlinear-ODE + reset | IF, RF | × |
| Digital spiking neuron [7]-[9] | Discrete state | NA IF | ○ |
| Resonate-and-fire-type DSN ([10] and in this paper) | Discrete state | IF, RF | ○ |

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 392–400, 2009.

(b) In the proposed model, the parameter which characterizes the dynamics is a pattern of the wirings among the shift registers. Using a reconfigurable devices, we can adjust the parameter values on-chip as shown in Table 1. By contrast, in analog spiking neuron models, typical parameter values are characteristics of analog device (e.g., nonlinear register). It is very hard to adjust such parameter values dynamically on-chip as shown in Table 1.

## 2 Resonate-and-Fire-Type Digital Spiking Neuron

In this section, we present a resonate-and-fire-type digital spiking neuron. As shown in Fig.1(a), the neuron consists of four parts: $N$ pieces of $u$-cells that are index'ed by $i \in \{0, 1, \ldots, N-1\}$; $M$ pieces of $v$-cells that are indexed by $j \in \{0, 1, \ldots, M-1\}$; one-way wirings from the $u$-cells to the $v$-cells; and a shift direction controller. The $u$-cells and the $v$-cells correspond to a recovery variable and a membrane potential of the Izhikevich's simple model [1], respectively. Each $u$-cell has a binary state $U_i(t) \in \{0, 1\}$, where $t \in [0, \infty)$ is a continuous time. We assume that one $u$-cell has a state "1" and the other $u$-cells have states "0". Then we can introduce the integer state $u(t) = i$ if $U_i(t) = 1$, where $u(t) \in \{0, 1, \ldots, N-1\}$. Similarly, each $v$-cell has a binary state $V_j(t) \in \{0, 1\}$, one $v$-cell has a state "1", and the other $v$-cells have states "0". Then we can introduce the integer state $v(t) = j$ if $V_j(t) = 1$, where $v(t) \in \{0, 1, \ldots, M-1\}$. The stimulation input $s(t) \in \{0, 1\}$ and the internal clock $c(t) \in \{0, 1\}$ change the states $(v, u)$ of the neuron as explained below (see also Fig.1(b)).

*The stimulation input* : The stimulation input $s(t)$ is applied to the $v$-cells only and shifts the state $v$ upward. In this paper, we focus on a periodic input with period $d \in [0, \infty)$ as shown in Fig.1(b). We can describe the time evolution of the state $v$ by the stimulation input $s(t)$ $(c(t) \equiv 0)$ as follows.

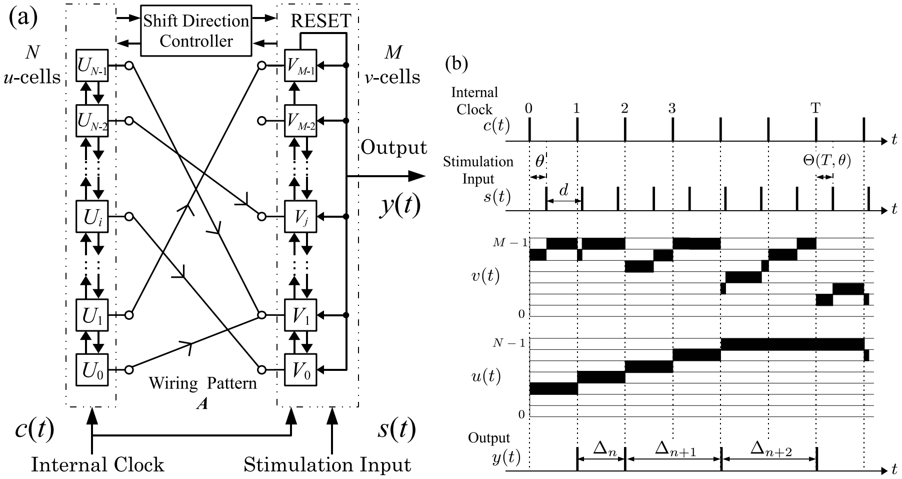


**Fig. 1.** (a)The resonate-and-fire-type digital spiking neuron. (b)The neuron dynamics.

**Table 2.** Control rules of the shift direction controller. ↑:shift up, ↓:shift down, $M_c := \left\lfloor \frac{M-1}{2} \right\rfloor$.

| Subspaces | $u$-cells | $v$-cells |
|---|---|---|
| $\mathbf{S}_0 := \{(v,u) \mid v = M_c, u = f(v)\}$ | fixed | fixed |
| $\mathbf{S}_1 := \{(v,u) \mid v > M_c, u \geq f(v)\}$ | ↑ | ↓ |
| $\mathbf{S}_2 := \{(v,u) \mid v \leq M_c, u > f(v)\}$ | ↓ | ↓ |
| $\mathbf{S}_3 := \{(v,u) \mid v < M_c, u \leq f(v)\}$ | ↓ | ↑ |
| $\mathbf{S}_4 := \{(v,u) \mid v \geq M_c, u < f(v)\}$ | ↑ | ↑ |

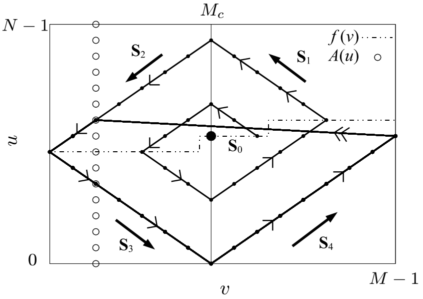


**Fig. 2.** Shift directions and typical orbit (with no stimulation input $s(t)$)

$$v(t+d) = \begin{cases} v(t)+1 & \text{if } v(t) \neq M-1. \\ v(t) & \text{otherwise.} \end{cases} \tag{1}$$

*The internal clock* : The internal clock $c(t)$, which has the normalized period 1 (see Fig.1(b)), is applied to the both cells and shifts the states $(v,u)$. The shift direction is decided by the shift direction controller as shown in Fig.1(a). The controller rules the shift direction based on a function $f(v)$ as shown in Table 2. In this paper we use the following function $f(v)$.

$$f(v) = \begin{cases} a_1 & \text{if } v < \left\lfloor \frac{M}{3} \right\rfloor, \\ f^*\left(\left\lfloor b\left(v - \left\lfloor \frac{M-1}{2} \right\rfloor\right) + a_2 \right\rfloor\right) & \text{if } \left\lfloor \frac{M}{3} \right\rfloor \leq v < 2\left\lfloor \frac{M}{3} \right\rfloor, \\ a_3 & \text{otherwise,} \end{cases}$$
$$f : \{0,1,\ldots,M-1\} \to \{-1,0,\ldots,N\},$$

where $a_i \in \{-1,0,\ldots,N\}, b \in [-N,N]$ and $f^*(v) = (|v+1| - |v-N| + N - 1)/2$. For convenience, we use a parameter vector $\boldsymbol{f} = (a_1,a_2,a_3,b)$. As shown in Fig.2, the entire state space $\{(v,u) \mid 0 \leq v < M, 0 \leq u < N\}$ is divided into the subspaces $\{\mathbf{S}_0,\ldots,\mathbf{S}_4\}$. The control rule realizes a rotation dynamics (resonation-like dynamics) as shown in Fig.2.

When the state $v$ reaches $M-1$ (firing threshold) and an internal clock $c(t) = 1$ arrives, the state $v$ is reset to a value $A(u)$ which is decided by the pattern of the wirings. To describe this we introduce a wiring function $A(u), A : \{0,1,\ldots,M-1\} \to \{0,1,\ldots,N-1\}$. $A(i) = j$ means that a cell $U_i$ is connected to a cell $V_j$ (see Fig.1(a)). We represent the wiring function $A(u)$ by the following parameter vector

$$\boldsymbol{A} = (A(0), A(1), \ldots, A(M-1)).$$

And we refer to the parameter vector $\boldsymbol{A}$ as the wiring pattern. An example of $A(u)$ is in Fig.2. At the reset moment, a firing spike $y(t) = 1$ is generated as shown in Fig.1(b). In order to formulate the neuron dynamics we introduce the following functions.

$$x(v,u) := \begin{cases} 1 & \text{if } (v,u) \in \mathbf{S}_3 \cup \mathbf{S}_4. \\ -1 & \text{if } (v,u) \in \mathbf{S}_1 \cup \mathbf{S}_2, v \neq 0. \\ 0 & \text{otherwise.} \end{cases}$$

$$z(v,u) := \begin{cases} 1 & \text{if } (v,u) \in \mathbf{S}_1 \cup \mathbf{S}_4, u \neq N-1. \\ -1 & \text{if } (v,u) \in \mathbf{S}_2 \cup \mathbf{S}_3, u \neq 0. \\ 0 & \text{otherwise.} \end{cases}$$

Using above, we can describe the time evolution of the states $(v,u)$ by the internal clock $c(t)$ $(s(t) \equiv 0)$ as follows.

$$v(t+1) = \begin{cases} v(t) + x(v(t), u(t)) & \text{if } v(t) \neq M-1. \\ A(u(t)) & \text{otherwise.} \end{cases} \tag{2}$$

$$u(t+1) = u(t) + z(v(t), u(t)) \tag{3}$$

The neuron accepts both the stimulation input $s(t)$ and the internal clock $c(t)$, and then the dynamics is described by the system of Equations (1), (2) and (3). Repeating the rotation and firing dynamics, the neuron generates a spike-train $y(t)$. Examples of the orbit on a phase plane are in Fig.3.

# 3 Return Map and Analysis

## 3.1 Return Map

In this section we derive a return map to analyze the neuron. As shown in Fig.1(b), we define $t = 0$ as the time when the internal clock $c(t)$ is applied first, and a variable $\theta \in (0, d] =: \mathbf{\Theta}$ as the initial phase of the stimulation input $s(t)$. Let $t_1 > 0$ and $t_2 > t_1$. We define a function $I_{pt}(t_1, t_2)$ as the number of the spikes of the stimulation input $s(t)$ during $t_1 < t \leq t_2$ as follows.

$$I_{pt}(t_1,t_2) = \begin{cases} \lfloor (t_2-\theta)/d \rfloor - \lfloor (t_1-\theta)/d \rfloor & \text{if } t_1 \geq \theta. \\ \lfloor (t_2-\theta)/d \rfloor + 1 & \text{if } t_2 \geq \theta > t_1. \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Additionally, we define a variable $v'(t)$ as follows.

$$v'(t) := v(t) + I_{pt}(t, t+1).$$

Then we can derive a discrete time map which represents the dynamics of the neuron for a discrete time $\tau \in \{0, 1, 2, \ldots\} =: \mathbf{T}$ as follows.

$$v(\tau+1) = \begin{cases} v'(\tau) + x(v'(\tau), u(\tau)) & \text{if } \; v'(\tau) < M-1. \\ A(u(\tau)) & \text{otherwise.} \end{cases} \tag{5}$$

$$u(\tau+1) = u(\tau) + z(v'(\tau), u(\tau)). \tag{6}$$

Using Equations (5) and (6), the states $(v,u)$ are determined uniquely for all $\tau$.

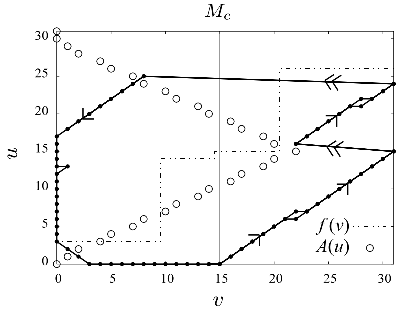


**Fig. 3.** Phase plane and typical orbit. Parameters are $M = N = 32$, $\boldsymbol{f} = (3, 15, 26, 0.1)$ and $A(u) = -|1.5(u - 15)| + (M - 10)$.

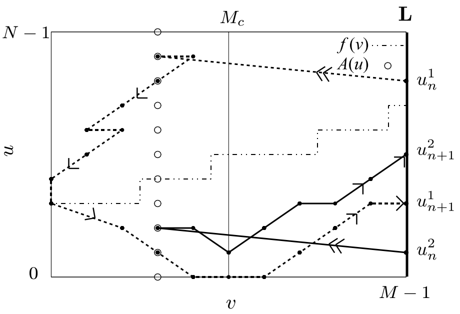


**Fig. 4.** Definition of the return map. Each $u_n^k$ map to $u_n^{k+1}$.

Next, we derive a return map on the firing threshold $\mathbf{L} = \{(v, u) \mid v = M - 1, 0 \le u < N\}$ as shown in Fig.4. Let a point on $\mathbf{L}$ be represented by its $u$-coordinate. When the states $(v, u)$ are on the threshold $\mathbf{L}$ we set $\tau = 0$, hence

$$\text{if} \quad \left\{ \begin{aligned} & u(\tau - 1) \in \mathbf{L} \\ & v(\tau) = A(u) \\ & u(\tau) = u + z(M - 1, u) \end{aligned} \right\} \quad \text{then} \quad \tau <= 0,$$

where "<=" represents "is reset to." The return map consists of two variables: the first one is $u_n \in \mathbf{L}$ which means the starting point of $u(\tau)$ on $\mathbf{L}$; the second one is $\theta_n \in \boldsymbol{\Theta}$ which means a phase of the stimulation input $s(t)$ with respect to the starting moment $\tau = 0$, where $n \in \{0, 1, 2, \ldots\}$. Hence the return map is two-dimensional. And we denote an inter spike interval (ISI) by $\Delta_n$ (see Fig.1(b)). The ISI $\Delta_n$ is given by

$$\begin{gathered} \Delta_n = H(u_n, \theta_n) = \min\{\tau + 1 \mid \tau \in \mathbf{T}, v'(\tau) \ge M - 1\}, \\ H : \mathbf{L} \times \boldsymbol{\Theta} \to \{1, 2, \ldots\}, \end{gathered} \tag{7}$$

where $v'(\tau)$ is decided by $(u_n, \theta_n)$, so $\Delta_n$ is a function of $(u_n, \theta_n)$. And we define a function $\Theta(\tau, \theta)$ as follows.

$$\Theta(\tau, \theta) = \min\{\theta + ld - \tau \mid l, \tau \in \mathbf{T}, \theta + ld > \tau\}. \tag{8}$$

$\Theta(\tau, \theta)$ denotes the phase of the stimulation input $s(t)$ with respect to the moment $\tau$ as shown in Fig.1(b). Finally we can derive a return map as follows.

$$\begin{gathered} u_{n+1} = F(u_n, \theta_n) = u(\Delta_n - 1), \\ \theta_{n+1} = G(u_n, \theta_n) = \Theta(\Delta_n, \theta_n), \\ F : \mathbf{L} \times \boldsymbol{\Theta} \to \mathbf{L}, \quad G : \mathbf{L} \times \boldsymbol{\Theta} \to \boldsymbol{\Theta}. \end{gathered} \tag{9}$$

As a result, the dynamics of the neuron is described by the return map in Equation (9) and the function $H(u_n, \theta_n)$ which gives the ISI $\Delta_n$ without approximations. Examples of the projection $F(u_n, \theta_n)$ of the return map (9) are shown in Fig.5.

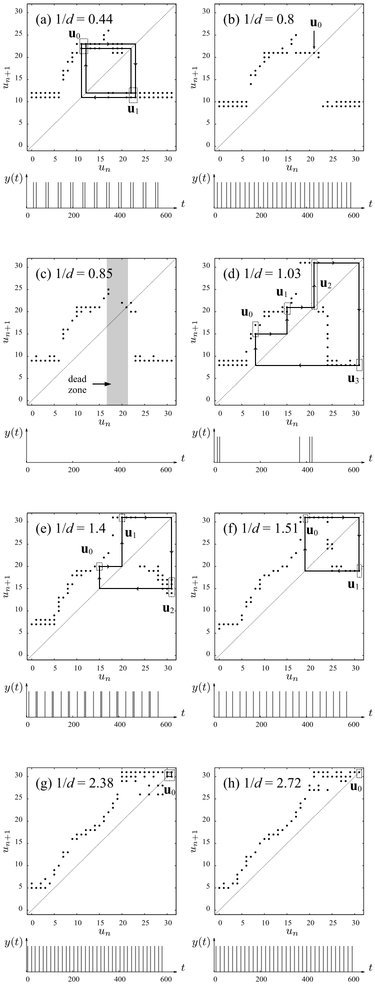


**Fig. 5.** Projection $F(u_n, \theta_n)$ of the return map with typical orbits and the output spike-train y(t)

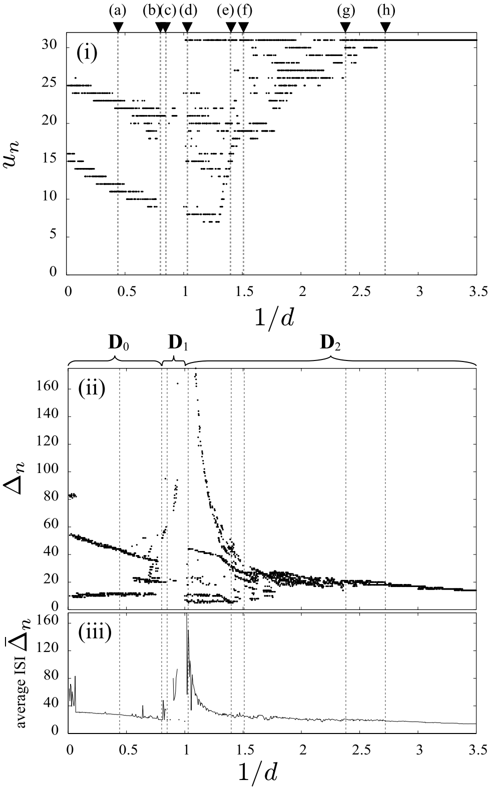


**Fig. 6.** (i) Bifurcation diagram of the state $u_n$ for the parameter $1/d$. Points (a)-(h) correspond to Fig.5(a)-(f), respectively. (ii)Characteristics of the ISI $\Delta_n$. (iii)Average of the ISI $\bar{\Delta}_n$.

## 3.2 Analysis

Using the ISI map in Equation (7) and the return map in Equations (9), we can analyze the bifurcation mechanisms and response characteristics of the neuron. The parameters which we use are in Fig.3. Fig.6 shows a bifurcation diagram of the state $u_n$ on the firing threshold **L** and characteristics of the ISI for the frequency $1/d$ of the stimulation input $s(t)$. And we introduce the following definitions.

**Definition 1.** *A set* $\mathbf{u}_0$ *is said to be a fixed set if* $\mathbf{u}_0$ *is the minimum set such that* $u_n \in \mathbf{u}_0$ *for all* $n \geq 1$.

**Definition 2.** *Let $r \geq 2$ be a positive integer. Sets* $(\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{r-1})$ *are said to be period-r sets if they are the minimum disjoint sets such that* $u_n \in \mathbf{u}_{n+q \pmod r}$ *for all* $n \geq 1$ *and for a fixed integer* $q \in \{0, 1, \ldots, r-1\}$.

Next, we explain the characteristics of the neuron in each regions $\{\mathbf{D}_0, \mathbf{D}_1, \mathbf{D}_2\}$ as shown in Fig.6(ii).

***Period-2 sets* in $\mathbf{D_0}$ to *fixed set.* in $\mathbf{D_0}$.** In Fig.5(a), there are *period-2 sets* $(\mathbf{u}_0, \mathbf{u}_1)$ so the output $y(t)$ is 2-periodic. Increasing the input frequency $1/d$, a left branch of the map is dropping. In Fig.5(b), the branch has a crossing set with diagonal line $u_{n+1} = u_n$. The set is a *fixed set* $\mathbf{u}_0$ so the output $y(t)$ is 1-periodic (see Fig.5(b)). In this region $\mathbf{D}_0$, the average of the ISI $\bar{\Delta}_n$ is decreasing as the input frequency $1/d$ increases.

***Fixed set* in $\mathbf{D_0}$ to dead zone in $\mathbf{D_1}$.** In Fig.5(c) shaded area is drawn. In this area the states $(v, u)$ hardly ever returns to the firing threshold $\mathbf{L}$. Then the neuron shows subthreshold oscillation and the output $y(t)$ is not observed. In the subthreshold oscillation case, the average of the ISI $\bar{\Delta}_n$ is not defined. We refer to the area "dead zone".

**Dead zone in $\mathbf{D_1}$ to *period-4 sets* in $\mathbf{D_2}$.** When the input frequency $1/d$ is increasing and enters into the region $\mathbf{D}_2$, the dead zone disappears. In Fig.5(d) there are *period-4 sets* $(\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$, so the output $y(t)$ is 4-periodic. Also, in the case of Fig.5(d), the average of the ISI $\bar{\Delta}$ is large because the states $(v, u)$ rotates long time under the threshold.

**Periodic sets in $\mathbf{D_2}$.** In Fig.5(e) the *period-4 sets* $(\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$ turns into *period-3 sets* $(\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2)$, so the output $y(t)$ is 3-periodic. In Fig.5(f), *period-3 sets* $(\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2)$ turns into *period-2 sets* $(\mathbf{u}_0, \mathbf{u}_1)$. But the output $y(t)$ seems like 1-periodic because two ISIs corresponding to the *period*-2 sets are very close as shown in Fig.6(ii). Increasing the input frequency $1/d$ further, *period-2 sets* shrinks and turns into *fixed set* $\mathbf{u}_0$ (see

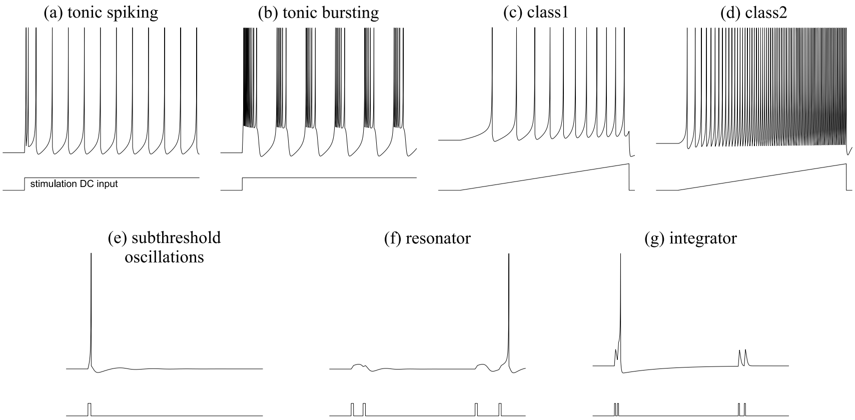


**Fig. 7.** Some neurocomputaional properties in the simple model [2]

**Table 3.** Comparison of properties of the spiking neuron models.(a)-(g) are properties corresponding to Fig.7(a)-(g), respectively. BI: biophysically meaningful, OL: on-chip learning.

| Models | (a) | (b) | (c) | (d) | (e) | (f) | (g) | BI | OL |
|---|---|---|---|---|---|---|---|---|---|
| Integrate-and-fire | ○ | × | ○ | × | × | × | × | × | × |
| Resonate-and-fire | ○ | × | ○ | ○ | ○ | ○ | ○ | × | × |
| FitzHugh-Nagumo | ○ | × | × | ○ | ○ | ○ | × | × | × |
| Moris-Lecar | ○ | × | ○ | ○ | ○ | ○ | ○ | ○ | × |
| Izhikevich | ○ | ○ | ○ | ○ | ○ | ○ | ○ | × | × |
| **The neuron in this paper** | ○ | ○ | ○ | ○ | ○ | ○ | ○ | × | ○ |

Fig.5(f)-(h)). In the region $\mathbf{D}_2$, a right branch of the map is going up as the input frequency $1/d$ increases. Therefore, the number of the period and the average of the ISI $\bar{\Delta}_n$ is almost decreasing.

**Summary of neural behaviors.** We have analyzed other bifurcation phenomena of the neuron and confirmed that the neuron can exhibit various phenomena. We summarize comparisons of the proposed neuron with other models in Table 3. As shown in this table, the proposed model can exhibit as much neural-behaviors as other models.

## 4 Conclusions

In this paper we presented a generalized model of a resonate-and-fire-type digital spiking neuron. And using a hybrid return map we analyzed four types of the bifurcation phenomena. As a result, we clarified that the generalized model can exhibits richer dynamics than the old model as shown in Table 3. Future problems include: (a) FPGA implementation of the neuron, (b) proposing a learning algorithm, and (c) development of network of the proposed neurons.

**Acknowledgments.** The authors would like to thank Professor Toshimitsu Ushio of Osaka University for valuable discussions. This work is partially supported by the Center of Excellence for Founding Ambient Information Society Infrastructure; Osaka University, Japan; KAKENHI (21700253); and TAF.

## References

1. Izhikevich, E.M.: Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting. The MIT press, Cambridge (2007)
2. Izhikevich, E.M.: Which model to use for cortical spiking neurons? IEEE Transactions on Neural Networks 15, 1063–1070 (2004)
3. Maass, W., Bishop, C.: Pulsed neural networks. Bradford Book (1999)
4. Hamanaka, H., Torikai, H., Saito, T.: Quantized spiking neuron with a/d conversion functions. IEEE Transactions on Circuits and Systems II: Express Briefs 53(10), 1049–1053 (2006)

5. Mitsubori, K., Saito, T.: Dependent switched capacitor chaos generator and its synchronization. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. 44(12), 1122–1128 (1997)
6. Matsuoka, Y., Hasegawa, T., Saito, T.: Chaotic spike-train with line-like spectrum. IEICE Trans. Fundamentals E92-A, 1142–1147 (2009)
7. Torikai, H., Funew, A., Saito, T.: Digital spiking neuron and its learning for approximation of various spike-trains. Neural Networks 21(2-3), 140–149 (2008)
8. Hashimoto, S., Torikai, H.: A novel hybrid spiking neuron: response analysis and learning potential. In: Proc. ICONIP (2008)
9. Hashimoto, S., Torikai, H.: Bifurcation analysis of a reconfigurable hybrid spiking neuron and its novel online learning algorithm. In: Proc. IEEE-INNS/IJCNN (2009)
10. Hishiki, T., Torikai, H.: A novel resonate-and-fire-type digital spiking neuron and its bifurcation analysis. In: Proc. NOLTA (accepted, 2009)

# Strange Responses to Fluctuating Inputs in the Hindmarsh-Rose Neurons

Ryosuke Hosaka[1,2], Yutaka Sakai[3], and Kazuyuki Aihara[1,4]

[1] ERATO Aihara Complexity Modelling Project, JST, Tokyo 113-8656, Japan
[2] RIKEN Brain Science Institute, Wako 351-0198, Japan
[3] Tamagawa University Brain Science Institute, Tokyo 194-8610, Japan
[4] Institute of Industrial Science, University of Tokyo, Tokyo 113-8656, Japan

**Abstract.** A Input-Output relationship of a neuron is often used to characterize the function of the neuron. In that case, inputs are assumed to be constant, and neuronal responses to fluctuated inputs are less understood. A recent work reported a strange response of the Hodgkin-Huxley neuron to the fluctuated inputs that an irregularity of spike trains is inversely proportional to an input irregularity. In this paper, we investigated an origin of the strange response by using the Hindmarsh-Rose neuron. We provided the parameter regions for bifurcations and confirmed that the Hindmarsh-Rose neuron reproduces the strange response in dynamics of Saddle-Node and Subcritical Hopf bifurcations. In both bifurcation cases, the Hindmarsh-Rose neuron shows a bistability of resting potential and repetitive firing. This indicates that the bistability is the origin of the strange input-output relationship.

**Keywords:** Hindmarsh-Rose neuron, interspike interval, subthreshold oscillation, bistability.

## 1 Introduction

A fundamental function of a neuron is to transform inputs to outputs. A neuron therefore can be characterized by its Input-Output (I-O) transformation. The inputs to the neurons are in general synaptic currents and the outputs are neuronal spikes. A classification of neurons by using a frequency-Input (f-I) function is proposed by Hodgkin [1]. The Phase response curve also characterizes the neuronal I-O functions. In these cases, the inputs are constant currents.

A cortical neuron generates irregular spike trains including highly variable intervals, telling us a fact that a cortical neuron receives highly fluctuating inputs. The fluctuating input can be realized by balancing excitatory and inhibitory synaptic inputs [2,3]. This balanced input has several advantages on information processing: it make an irregularity of neuronal firing remain approximately constant regardless of the firing rate [4] and has a capacity for gating multiple signals [5]. So a neuronal response to the fluctuating input is a crucial function.

To clarify the I-O functions of neurons to fluctuating inputs, several works have reported the responses of the neurons to the fluctuated inputs and reported reactive differences among the neuron models [6,7,8,9,10]. Recently, an

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 401–408, 2009.

interesting phenomenon was reported: the variability of output spike trains of the Hodgkin-Huxley neuron model (HH) decreases as the input variance increases (Shape of this relationship is almost the same with $\overline{T}/\tau \geq 20$ of Fig.3(SubAH)) [11]. This inverse relationship of input and output variances unmatches with our intuition, so we call it "strange response" here. In fact, the input-output relationship for a leaky integrate-and-fire neuron model (LIF) is not strange but proportional. The authors concluded the paper by providing a possible underlying mechanisms, that the strange response of HH might come from the subthreshold oscillation of the membrane potential.

Although the finding is important and fundamental, further analysis is necessary, because the comparison has been done with the models whose dynamics are largely different each other: HH is based on the ion conductances and written by more than four-dimensional differential equations; LIF ignores ion channels and mimics only sub-threshold membrane potentials. There are a lot of differences between HH and LIF, a complexity of dynamics, the number of variables, the number of parameters, and so on. Moreover, HH is too complicated to find the origin of the strange response. Therefore, we cannot acknowledge that, as the authors concluded, the subthreshold oscillation is the origin of the strange response. Other components might cause the strange response. The purpose of this paper is to find the origin of the strange response.

The major differences between HH and LIF are two points, the subthreshold oscillation of membrane potential and a bistability of attractors, resting potential and repetitive firing. HH has both, LIF has neither. To clarify the origin of the strange response, the subthreshold oscillation and bistability should be separated.

We first showed that the two-dimensional Hindmarsh-Rose model (HR) is capable to separate the subthreshold oscillation and bistability. HR is a neuron model that is described by only two-variables and has much fewer parameters than HH. We then show that the origin of the strange response is not the subthreshold oscillation but the bistability.

## 2 Hindmarsh-Rose Model

HR is described as follows:

$$\dot{x} = x - x^3/3 - y + I(t) \tag{1}$$

$$\tau\dot{y} = (x^2 + dx + a)/b - y \tag{2}$$

where $\dot{x}$ represents a temporal derivative of $x$ and $\tau := 3^2/b$ corresponds to a membrane time constant. One of the fixed points of HR mimics the resting potential. It becomes unstable through one of four types of bifurcation forms: Saddle-Node bifurcation on invariant circle (SNonIC), Saddle-Node bifurcation not on invariant circle (SNnoIC), supercritical Andronov-Hopf bifurcation (SupAH), and subcritical Andronov-Hopf bifurcation (SubAH) [12]. These four bifurcations respectively correspond to different combinations of the subthreshold

**Table 1.** Four types of behaviors and their corresponding bifurcations

| | subthreshold oscillation | |
|---|---|---|
| | no | yes |
| monostable | saddle-node on invariant circle<br>(SNonIC) | supercritical Andronov-Hopf<br>(SupAH) |
| bistable | saddle-node not on invariant circle<br>(SNnoIC) | subcritical Andronov-Hopf<br>(SubAH) |

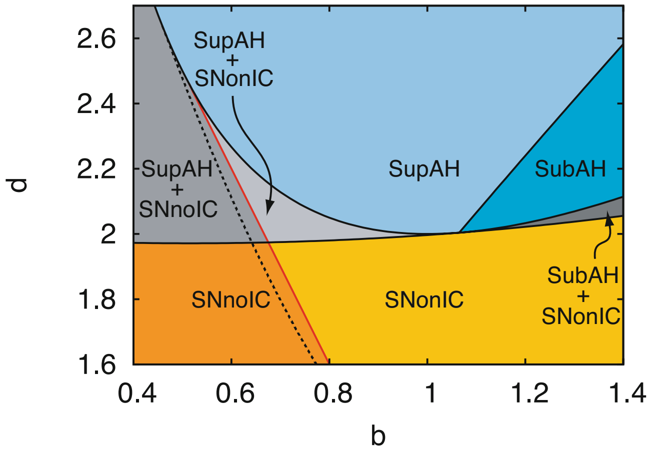


**Fig. 1.** Parameter regions for the bifurcations. The parameters in the three gray regions are not appropriate for neuron models because two bifurcations coexist.

oscillation and bistability. The correspondence is summarized in Table 1 [13]. HR can exhibit all the four bifurcations by operating two parameters $(b, d)$ [14,12]. The parameter region for each bifurcation is depicted in Fig. 1. We provided derivations of parameter regions in Appendix A. The following $(b, d)$ was used as the typical parameter value for each bifurcation: (1.0,1.8) for SNonIC, (0.6,1.8) for SNnoIC, (1.0,2.2) for SupAH, (1.3,2.2) for SubAH. The frequency-current relationship for them is given by Fig. 2. The input parameter region for the bistability is indicated by the grey color in the figure.

The inward current to a cell body, $I(t)$ in Eq.(1), is described by the form

$$I(t) = \mu + \sigma\xi(t), \tag{3}$$

where $\xi(t)$ is white Gaussian noise. The parameters $\mu$ and $\sigma$ control mean and fluctuation of inputs, respectively. This fluctuated input is based on the following assumption. A cortical neuron receives thousands of synaptic contacts. If incoming inputs through synapses were assumed to be independent, the sum of a large number of the independent excitatory and inhibitory inputs can be approximated to be a uncorrelated fluctuation [2].

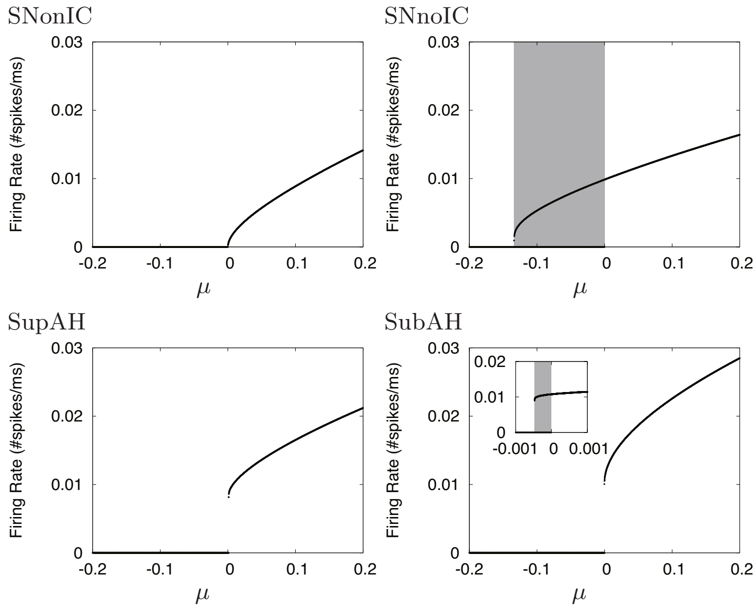


**Fig. 2.** Frequency-current relationship for a constant input current, $I(t) = \mu$. The grey regions on the SNnoIC and SubAH indicate the bistable conditions. Parameter $a$ is 0.073705 for SNonIC, -0.126226 for SNnoIC, 0.521833 for SupAH, and 0.319832 for SubAH, to make one of bifurcations occur at $\mu = 0$.

Output spike trains were evaluated by two statistics of interspike intervals (ISIs): the mean ISI ($\overline{T}$) and the coefficient of variation (Cv), defined respectively as

$$\overline{T} = \frac{1}{n}\sum_{i=1}^{n} T_i, \tag{4}$$

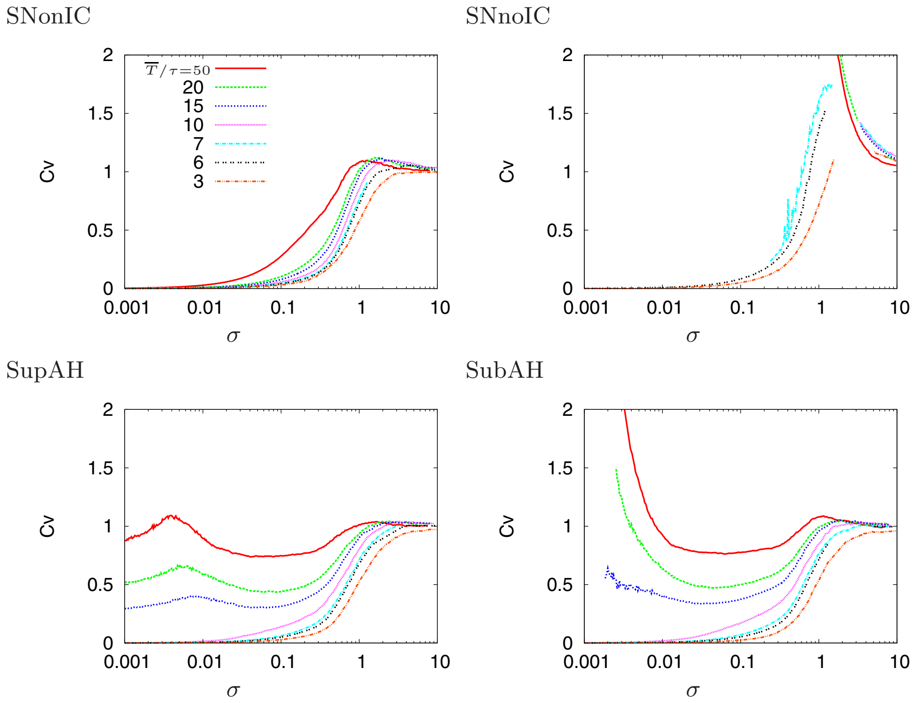
$$\mathrm{Cv} = \sqrt{\overline{(T_i - \overline{T})^2}}/\overline{T}, \tag{5}$$

where $T_i$ represents an ISI. Cv evaluates an irregularity of the spike trains. If the spike train is completely regular, that is, all ISIs are constant, Cv corresponds to 0. If the spike train is completely random, meaning Poisson process, Cv corresponds to 1. Because Cv is a dimensionless quantity, we can directly compare Cv for models. By contrast, $\overline{T}$ is not dimensionless value. We therefore use the ratio of $\overline{T}$ to membrane time constant, $\overline{T}/\tau$, for comparison.

## 3 Response of HR to Fluctuated Inputs

### 3.1 Does HR Show the Strange Responses?

We firstly confirmed if HR reproduces the strange input-output relationship as the same with the HH. Figure 3 shows the dependence of Cv on input variance


SNonIC

SNnoIC

SupAH

SubAH

**Fig. 3.** Relationship between input variance $\sigma$ and output variance Cv

$\sigma$ at each bifurcation conditions, where input mean $\mu$ was altered so that $\overline{T}/\tau$ should be constant.

HR with SNonIC exhibits neither the subthreshold oscillation nor bistability. This is the same as LIF. Cv of this model, depicted in Fig.3(SNonIC), increases as $\sigma$ increases, meaning that irregularity of output spike trains is proportional to the input variance $\sigma$. This is true regardless of $\overline{T}/\tau$. The similar orbit of Cv was observed in LIF [11].

HR with SubAH exhibits both the subthreshold oscillation and bistability, as the same with HH. As shown in Fig.3(SubAH), in the case $\overline{T}/\tau \leq 10$, Cv of this model resembled that of SNonIC: Cv is proportional to input variance $\sigma$. On $\overline{T}/\tau > 20$, Cv takes on a different appearance: Cv is inversely proportional to input variance $\sigma$, especially in the range $0.001 < \sigma < 0.1$. Even the input variance is below 0.01, Cv exceeds 1 and decreases as $\sigma$ increases. This inverse relationship is almost the same as HH [11]. This means that HR is complex enough to reproduce the strange responses and that the subthreshold oscillation and bistability are candidates of the origin for the strange response.

### 3.2 The Origin of the Inverse Relationship: Subthreshold Oscillation or Bistability

We comfirmed that HR with SubAH shows the inverse input-output relashionship. Which is the origin of the strange response, the subthreshold oscillation or bistability? To address this issue, we calculated Cv of HR with SupAH and HR with SNnoIC. HR with SupAH shows the subthreshold oscillation, but does not show the bistability. In contrast, HR with SNnoIC shows only the bistability, it does not show the subthreshold oscillation.

Fig.3(SupAH) depicts the Cv of HR with SupAH. Orbits of Cv of $\overline{T}/\tau \leq 10$ are almost the same as that of HR with SubAH. However, the orbits of Cv of $\overline{T}/\tau \geq 15$ are different from that of HR with SubAH: below $\sigma \leq 0.1$, Cv values almost remain constant even the input variance increases. This indicates that the subthreshold oscillation is not an origin of the inverse input-output relationship.

Fig.3(SNnoIC) depicts the Cv of HR with SNnoIC. Cv of $\overline{T}/\tau = 3$ is almost the same as other bifurcations. In $\overline{T}/\tau \geq 10$, Cv values show the inverse relationship to input variance $\sigma$.

As above results showed, HR with SupAH does not reproduce the inverse input-output relationship, while HR with SNnoIC reproduce the inverse relationship. From these, we concluded that the subthreshold oscillation is not the origin of the strange response and that the bistability of the resting states and repetitive firing is the origin of the strange response.

## 4 Discussion

In the present study, we derivated the parameter regions of HR for each bifurcation and confirmed that HR can reproduce the strange input-output relationships in SNnoIC and SubAH conditions. This indicates that the origin of the strange input-output relationship is not the subthreshold oscillation but the bistability of the resting state and repetitive firing.

The bistability of resting and repetitive firing have been observed in biological neurons in the entorhinal cortex of the brain [15]. In this neuron, activity-dependent changes of a $Ca^{2+}$-sensitive cationic current plays a critical role. Althoght functional meaning of the strange response is not known, it might play some roles in the entorhinal cortex.

In this study, the bistability of attractors was realised by a intrinsic mechanism of HR. The bistability may be able to be realised by the Up/Down states of membrane potential [16]. In cortical neurons, the membrane potential stay around -65mV in Down states and -45mV in Up states. Firing probability in the Up state is much higher than the Down states. The neurons with Up/Down states are widely observed in neocortex. The Up/Down state can be regarded as the bistability of attractors, so the strange response may be seen in whole neocortex.

In this study, we focused on the inverse relationship observed in HH, and the other strange responses observed in HR were ignored: (a) In HR with SubAH, Cv of $\overline{T}/\tau \leq 15$ is invariant when input variance $\sigma$ is altered between 0.01 and 1;

(b) In HR with SNnoIC, Cv of $\overline{T}/\tau = 7$ increases beyond 1 and then approaches Cv=1. These are interesting phenomena. They should be investigated in future works.

## Acknowledgments

We thank Dr. S. Tsuji for fruitful discussions.

## References

1. Hodgkin, A.L.: Journal of Physiology 107, 165 (1948)
2. Tuckwell, H.C.: Introduction to theoretical neurobiology. Cambridge University Press, Cambridge (1988)
3. Shadlen, M.N., Newsome, W.T.: Current Opinion in Neurobiology 4, 569 (1994)
4. Miura, K., Tsubo, Y., Okada, M., Fukai, T.: Journal of Neuroscience 27, 13802 (2007)
5. Vogels, T.P., Abbott, L.F.: Nature Neuroscience 12, 483 (2009)
6. Brown, D., Feng, J., Feerick, S.: Physical Review Letters 82, 4731 (1999)
7. Hosaka, R., Sakai, Y., Ikeguchi, T., Yoshizawa, S.: Journal of the physical society of Japan 75, 124007 (2006)
8. Fourcaud-Trocme, N., Hansel, D., van Vreeswijk, C., Brunel, N.: Journal of Neuroscience 23, 11628 (2000)
9. Shinomoto, S., Tsubo, Y.: Physical Review E 64, 041910 (2001)
10. Schreiber, S., Samengo, L., Herz, A.V.M.: Journal of Neurophysiology 101, 2239 (2009)
11. Sakai, Y., Yamada, M., Yoshizawa, S.: Proceedings of International Joint Conference on Neural Networks 2, 1275 (2002)
12. Tsuji, S., et al.: International Journal of Bifurcation and Chaos 17, 985 (2007)
13. Izhikevich, E.M.: Dynamical systems in neuroscience: The Geometry of Excitability and Bursting. The MIT Press, Cambridge (2007)
14. Hindmarsh, J.L., Rose, R.M.: Nature 296, 162 (1982)
15. Egorov, A.V., et al.: Nature 420, 173 (2002)
16. Steriade, M., Nunez, A., Amzica, F.: Journal of Neuroscience 13, 2352 (1993)
17. Guckenheimer, J., Holmes, D.: Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields. Springer, Heidelberg (1983)

## A Bifurcation Conditions

### Condition for Andronov-Hopf bifurcation

A Jacobian matrix around fixed point $x_0$ is $\begin{pmatrix} 1 - x_0^2 & -1 \\ (2x_0 + d)/b\tau & -1/\tau \end{pmatrix}$, and its eigen equation becomes $\lambda^2 + \lambda\{1/\tau + x_0^2 - 1\} + (bx_0^2 - b + 2x_0 + d)/b\tau = 0$. Since eigen values of the Andronov-Hopf bifurcation (AH) point are complex conjugates, $1/\tau + x_0^2 - 1 = 0$ and $(bx_0^2 - b + 2x_0 + d)/b\tau > 0$. From them and $b\tau = c^2 > 0$, the condition for AH is

$$x_0 = \pm\sqrt{1 - 1/\tau}, \tag{6}$$

$$d > -bx_0^2 - 2x_0 + b. \tag{7}$$

We set $x_0 < 0$ in this study. From Eq.(6), $b < c^2$.

### Boundary between SupAH and SubAH

Let $f(x) := x - x^3/3$ and $g(x) := (x^2 + dx + a)/b$. A criterion to distinguish SupAH and SubAH is given by $D := \{g'(x_0) - \frac{1}{\tau}\}f'''(x_0) - f''(x_0)\{g''(x_0) - f''(x_0)\}$, $D > 0$ for SupAH and $D < 0$ for SubAH [17]. Thus, $D = 0$ gives a boundary between SupAH and SubAH:

$$d = b/\tau + 2bx_0^2. \tag{8}$$

### Condition for Saddle-Node bifurcation

Since the fixed point $(x_f, f(x_f))$ corresponds to the crossing point of $f(x)$ and $g(x)$, $x_f$ satisfies

$$x_f - x_f^3/3 = (x_f^2 + dx_f + a)/b. \tag{9}$$

For saddle-node bifurcations (SN), Eq.(9) has one independent root and one multiple root, indicates that Eq.(9) has two independent inflection points. The inflection points are solutions of first-order derivative of Eq.(9),

$$bx_f^2 + 2x_f + d - b = 0, \tag{10}$$

and its solution is $x_f = \{-1 \pm \sqrt{1 - b(d - b)}\}/b$. Eq.(10) must not have multiple root, results in $1 - b(d - b) > 0$. This gives a condition for SN:

$$d < 1/b + b. \tag{11}$$

### Boundary between SNonIC and SNnoIC

If $\tau = \infty$, we can calculate the boundary between SNonIC and SNnoIC analytically (the red line in Fig.1). Given the extremal values of $f(x)$ as $\big(x_{\text{fex1}}, f(x_{\text{fex1}})\big)$ and $\big(x_{\text{fex2}}, f(x_{\text{fex2}})\big)$ where $f(x_{\text{fex1}}) < f(x_{\text{fex2}})$, the boundary satisfies

$$f(x_{\text{SN}}) = f(x_{\text{fex2}}) \tag{12}$$

where $\big(x_{\text{SN}}, f(x_{\text{SN}})\big)$ is SN point. In this case, $x_{\text{SN}} = \big\{ -1 - \sqrt{1 - b(d - b)}\big\}/b$ and $f(x_{\text{fex2}}) = 2/3$.

Otherwise, the following numerical procedure gives a rough boundary (the dashed line in Fig.1, if $\tau = 3^2/b$). Given a reunion point $\big(x_{\text{re}}, f(x_{\text{re}})\big)$ of $f(x)$ and the limit cycle, the boundary satisfies

$$f(x_{\text{SN}}) = f(x_{\text{re}}). \tag{13}$$

# Evaluation of Color Constancy Vision Algorithm for Mobile Robots

Yasunori Takemura and Kazuo Ishii

Kyushu Institute of Technogy, Graduate School of Life Science and System Engineering,
Brain Science and System Engineering, 2-4 Hibikino Wakamatsu Kitakyushu Fukuoka, Japan
{takemura-yasunori@edu., ishii@}brain.kyutech.ac.jp

**Abstract.** One of important subjects for mobile robots is the vision based decision making system, where the color constancy is big problem for robots which use color property to recognize environments. We have been working on color constancy vision algorithms using bio-inspired information processing as creatures can recognize color and shape of objects even if there exits a large change of light conditions in outdoor environments. In this paper, we evaluate the performances of color recognition using bio-inspired processing algorithms such as Self-Organizing Map (SOM), modular network SOM (mnSOM) and Neural Gas (NG). The experimental results in various light conditions are discussed.

**Keywords:** color constancy, robot vision, Self-Organizing Map, Neural Gas.

## 1 Introduction

One of important subjects for most of mobile robots is the vision based decision-making system, where the perceptual constancy is big problem for robots which use vision to recognize environments. There are several types of perceptual constancies: shape constancy, size constancy, color constancy, lightness constancy, distance constancy, and location constancy. The color constancy is a feature of the human color perception system which ensures that the perceived color of objects remains relatively constant under varying illumination conditions. As the color constancy is not realized, the working space of robots is limited and robots often miss to detect target objects. In order to realize robust robot system which works outdoor environments and in various and varying lighting conditions, a system to keep color constancy is needed by adapting vision system to various color profiles of working environments. As the target robots system, we use RoboCup robots which move around using omni-vision camera indoor, because the lighting condition varies from hour to hour, weather also affects to color recognition, and a certain level of repeatability of lighting condition can be expected.

RoboCup is an international joint project to promote Artificial Intelligent (AI), robotics, and related fields. It is an attempt to foster AI and Intelligent robotics research by providing standard problems where a wide variety of technologies can be integrated and examined. In RoboCup, soccer game is selected as a main topic of research, aiming at innovations to be applied for socially significant problems and industries in the future [1].“Hibikino-Musashi” is a joint RoboCup middle-size league

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 409–419, 2009.

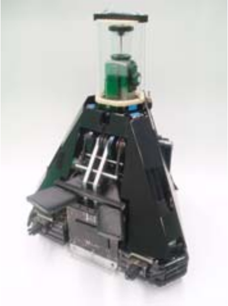


**Fig. 1.** “Musashi” robot includes an omni-directional platform, an omni-vision, and strong novel ball-kicking device, laptop PC and 90 [W] motor, omni-directional wheels and batteries

(MSL) soccer team in Kitakyushu Science and Research Park[2,3]. “Musashi” robot is developed based on the concept of the “omni-directional” and “modularity” concept (Fig.1) [4].

In this paper, we present the color recognition system based on color constancy algorithm using bio-inspired information technology. As the bio-inspired technology, we evaluate Multi-Layer Perceptron (MLP), Self-Organizing Map (SOM), modular network SOM (mnSOM), k-means, Neural Gas (NG). For examples, Self-Organizing Map (SOM) is an unsupervised learning algorithm that performs topology-preserving transformation from higher-dimensional vector data spaces to low map spaces. The SOM has become a powerful tool in many areas such as data mining, data analysis, data classification, and data visualization [5,6,7,8].

In the RoboCup, robots must detect a ball colored by orange, field by green and lines by white. Our current robot vision system uses two color models in YUV and HSV formats to detect target objects, and obtain different color images from both color models. Then, the images are binalized to detect target colors using a certain thresholds and added logically [9]. The important question is how the thresholds should be decided. In this paper, we introduce bio-inspired processing algorithms such as Self-Organizing Map (SOM), modular network SOM (mnSOM) and Neural Gas (NG) into the optimization of threshold parameters. The experimental results in various light conditions are discussed and evaluated.

## 2 Vision System of “Musashi” Robot

The vision system of the “Musashi” robot consists of an omni-directional mirror and an IEEE 1394 digital camera. Many RoboCup robots recently have strong kicking devices for high loop-shoots, so that the vision system should estimate the position of the ball in the air from single vision.

The obtained image from the omni-vision camera (Fig.2a) is in the YUV format and also converted into HSV format (Fig.2c). The upper and lower thresholds to extract target colors are set in both YUV and HSV formats, and the resulted image is given as the AND operation of both images. In the example of Fig.2, the blue region are extracted in YUV color space (Fig.2b) and HSV color space (Fig.2c) and the obtained image is in Fig.2d. It is shown that the blue goal is detected clearly using this method. In RoboCup, the robot should recognize three kinds of objects: the orange ball, the green field and the white lines. The orange ball is extracted as follows (1):

$$O = (V^o \cap H^o),\ V^o \in [V^o_{min}, V^o_{max}],\ H^o \in [H^o_{min}, H^o_{max}] \tag{1}$$

Let $O$, $V^o$and $H^o$ be the extracted orange region by AND operation, the orange region estimated from V value in YUV format, and the orange region from H value in HSV format, respectively. YUV format values and HSV format values are expressed by 8 bits numbers [0, 255]. Let $G$ and $W$ be the green region and the white line region, respectively. The following describes the process to extract the field region and line region.

$$G = (U^g \cap V^g),\ U^g \in [U^g_{min}, U^g_{max}], V^g \in [V^g_{min}, V^g_{max}] \tag{2}$$

$$W = Y^w,\ Y^w \in [Y^w_{min}, Y^w_{max}] \tag{3}$$

$Y$ and $U$ indicate the Y and U values in YUV, respectively. The upper suffix means the name of color, and the bottom means minimum (min) or maximum (max) values. This method improves the color extraction robustness and accuracy in variable lighting condition. We set the thresholds manually in the current system.

# 3 Color Constancy System Using Brain-Inspired Technology

## 3.1 Input/Output Data

In this paper, Bio-Inspired Algorithms are used for the recognition to the light environment condition. At first, we explain a basic input data and output data.

The basic four kinds of colors are prepared around the omi-directional camera. In this time, the input data $\boldsymbol{x}$ describes as follows:

$$x = (Y_g, U_g, V_g, Y_r, U_r, V_r, Y_b, U_b, V_b, Y_w, U_w, V_w) \tag{4}$$

The variable $Y_*$, $U_*$ and $V_*$ correspond to luminance information (Y) and color information (U and V) and take values between 0 – 255 originally and are normalized between -1.0 and 1.0. The bottom index describes green ($g$), red ($r$), blue ($b$) and white ($w$). This input data set is made from camera law data signal.

In this time, we need extract the colors. Output data needs the three kinds of the color thresholds: There are orange (ball), white (line) and green (fields), because the RoboCup Middle Size League (MSL) changes the rule which changes color goal (blue and yellow) to no color goal in this year. Orange is recognized with equation (1). $H_{\min}$ are always constant value. Therefore, $H_{\min}$ except the output data. Green is recognized with equation (2). $U_{\min}$ are always constant value. Therefore, $U_{\min}$ except the output data. White is recognized with equation (3). $Y_{\max}$ are also always constant value. Therefore $Y_{\max}$ also except the output data. In this time, output data set $y$ is described as follows:

$$y = (H^o_{\max}, V^o_{\min}, V^o_{\max}, V^g_{\max}, V^g_{\max}, Y^w_{\min}) \tag{5}$$

Upper index means orange (o), green (g), and white (w). All parameters are normalized -1.0 to 1.0 vectors. The proposed color constancy algorithms learn the mapping functions between the teaching vectors in (4) and (5). We suppose that the *x* in (4) expresses the fundamental environmental information and is used to decide the light condition. The *y* corresponds to the lighting condition.

### 3.2 Color Constancy Algorithm Using Self Organizing Map

In a Self-organizing map (SOM), the neurons are placed at the nodes of a lattice that is usually two-dimensional. Higher-dimensional maps are also possible but not as common. The neurons become selectively tuned to various input patterns or classes of input patterns in the course of a competitive learning process. The locations of the neurons so tuned (i.e., the winning neurons), neurons become ordered with respect to each other in such a way that a meaningful coordinate system for different input features is created over the lattice [10].

A SOM is therefore characterized by the formation of a topographic map of the input patterns in which the spatial locations (i.e., coordinates) of the neurons in the lattice are indicative of intrinsic statistical features contained in the input patterns, hence the name "Self-Organizing Map". In this time, the SOM algorithm is used on color detection algorithm.

SOM algorithm has the four processes: there are *Evaluative process*, *Competitive process*, *Cooperative process* and *Adaptive process*. Table I shows the variables used explanation of the color constancy algorithm of using batch type SOM. The learning data is defined by eq. (6).

$$\theta_i = [x_i \quad y_i] \tag{6}$$

(a) *Evaluative Process*

In evaluative process, all learning data sets are calculated the distance between each reference vector.

$$E_i^k = \left\| w^k - x_i \right\|^2 \tag{7}$$

(b) *Competitive process*

In competitive process, to find the best matching of the input vector *x* with the reference vector *w*, the best matching unit (BMU) (*k**) is defined that the minimum distance of eq. (7).

$$k_i^* = \arg_k \min E_i^k \tag{8}$$

(c) *Cooperative process*

In cooperative process, the winning units (best matching units) locate the center of a topological neighborhood of cooperating units. Let $d(k,k^*)$ denote the Euclidean distance between *k-th* module and $k^*$-th best matching unit. Then, we may assume that the topological neighborhood $\phi_i^k$ is an unimodal function of the distance $d(k, k^*)$, such that it satisfies two distinct requirement:

(i) The topological neighborhood $\phi_i^k$ is symmetric about the maximum point defined by $d(k, k^*) = 0$

(ii) The amplitude of the topological neighborhood $\phi_i^k$ decreases monotonically with increasing distance *d(k, k$^*$)*, decaying to zero for $d(k,k^*) \to \infty$; this is a necessary condition for convergence.

A typical choice of $\phi_i^k$ that satisfies these requirements are the Gaussian function (eq. (20)).

$$\phi_i^k = \exp\left(d(k,k_i^*)^2 / 2\sigma^2\right) \tag{9}$$

The parameter $\sigma$ is the "effective width" of the topological neighborhood. It is called neighbor radius. Its use also makes the SOM algorithm converge more quickly than a rectangular topological neighborhood world [11, 12, 13]. Another unique feature of the SOM algorithm is that the size of the topological neighborhood shrinks with time. This requirement is satisfied by making the neighbor radius $\sigma$ (eq. (10)) of the topological neighborhood function $\phi_i^k$ decrease with time. A popular choice for the dependence of $\sigma$ on discrete time *t* is the exponential decay described by [14, 15].

$$\sigma = \sigma_{\min} + (\sigma_{\max} - \sigma_{\min}) \exp(-t/\tau) \tag{10}$$

In batch type SOM, each unite learning rate is defined by $\psi_i^k$ which is normalized by summation of the $\phi_i^k$ (eq. (11))

$$\psi_i^k = \phi_i^k \Big/ \sum_{i'} \phi_{i'}^k \tag{11}$$

(d) *Adaptive process*

In the Kohonen's SOM, in adaptive process, all unit vectors are adjusted by using the update formula eq.(12)

$$w^k(t+1) = w^k(t) + \psi_i^k (x_i - w^k) \tag{12}$$

**Table 1.** Variables used in explanation of SOM algorithm

| Symbol | Quantity |
|---|---|
| $\theta$ | Learning data |
| $x$ | Input vector |
| $y$ | Output vector |
| $i$ | Index expressing classes (i = 1, .... , I) |
| $w$ | Reference vector |
| $k$ | Index expressing unit (k = 1,......, K) |
| $E$ | Distance between input vector and reference vector |
| $k^*$ | Best Matching Unit (BMU) |
| $\phi$ | Neigubor Function |
| $d(a,b)$ | Euclidean distance between a and b |
| $\psi$ | Learning late |
| $\sigma$ | Neighbor radius |
| $\tau$ | Time constant |

In this color constancy algorithm, learning data $\boldsymbol{\theta}$is described by eq.(17). So, unit vector has also input vectors and output vectors. In other words, reference vectors have also same dimensions of input and output vectors. In this time, we use the batch SOM, then the reference vectors ***w*** are updated by eq. (13).

$$w^{k}(t+1) = \sum_{i} \psi_{i}^{k} \theta_{i} \tag{13}$$

In the processing mode, input data form the camera put the evaluation process. And, on the competitive process, the robot finds the BMU, and this BMU output vector *y* is used for thresholds.

### 3.3 Color Constancy Algorithm Using Neural Gas

Neural Gas (NG) is network model of using vector quantization. NG is also similar in SOM some respects [16]. About one of the biggest different point, SOM has the relationship of neighborhood unit, on the other hand, NG doesn't have the relationship each unit. Each unit is learning independently, and there can move freely. Therefore, number of the unit is more decrease than SOM. Basically, the algorithm of NG learning process is same as SOM. NG has also 4 processes that is (a) evaluative process, (b) competitive process, (c) cooperative process and (d) adaptive process. In the (a), (d) processes are almost same with SOM algorithm. In the point of competitive process, the matrix of the Distance of input vectors and reference vectors are defined by eq. (14).

$$\boldsymbol{E}^{k} = \left(E_{1}^{k}, E_{2}^{k}, \ldots\ldots, E_{I}^{k}\right) \tag{14}$$

Then, the matrix of ***E*** was sorted in ascending sequence (eq.15). In this time, the number of the data(*i*) describes eq.(16). *j* (*j* = 1, ……, K) denote the index of eq.(15).

In the cooperative process, learning rate is calculated by eq.(17)

$$^{S}\boldsymbol{E}^{k} = sort(\boldsymbol{E}^{k}) \tag{15}$$

$$i = index(^{S}E^{k}(j)) \tag{16}$$

$$\phi_{i}^{^{S}\boldsymbol{E}^{k}(j)} = \exp\left( j \middle/ 2\sigma^{2} \right) \tag{17}$$

Equation of (17) substitute eq. (11), then, each units are learned by eq.(13). On the processing mode, the algorithm of processing mode is same as SOM.

### 3.4 Color Constancy Algorithm Using Modular Network SOM (mnSOM)

Modular network SOM (mnSOM) is extended to conventional SOM. The idea of mnSOM is very simple; each vector unit of a conventional SOM which is arrayed on a 2-dimentional lattice, is replaced by a function module of a neural network (e.g. Multi-layer perceptron (MLP), SOM, etc.) [17]. mnSOM inherits all other properties of the conventional SOM. The mnSOM strategy has several advantages. First, the application targets are widely expanded from fields involving just vectorized data to those dealing with more general classes of datasets relevant to functions, systems, time series and so on [18].

The learning process of mnSOM consists of 4 processes. Table II shows the functions and parameters of using mnSOM.

In our algorithm, the basic mnSOM does not have reference vectors. However, our algorithm has reference vectors same dimensions as input vector *x*, and neural network is used MLP which is learning under the Back-Propagation algorithm. In this time input is used input vector *x*, and teaching data is used output vector *y*.

(a) Evaluative process.
At first, input vector *x* is compare to the module's vector *u* (eq.(18)).

$$Em_j^m = \left\| u^m - x_j \right\|^2 \tag{18}$$

(b) Competitive process
The minimum of distance eq. (18) is decided as Best Matching Module (BMM).

$$m_j^* = \arg_m \min Em_j^k \tag{19}$$

(c) Cooperative process
By using each BMM, each neighbor hood module are given to the learning rate which is based on Gaussian function (eq.(20)).

$$\psi_j^m(t) = \phi(d(m,m_j^*)) / \sum_{j'=1}^{J} \phi(d(m,m_{j'}^*) \tag{20}$$

$$\phi(d(m,m_j^*);t) = \exp\left[-d(m,m_j^*)^2 / 2\sigma^2(t)\right] \tag{21}$$

(d) Adaptive process
By using the each learning rate, the MLP and reference vector are learned.

$$\Delta w^m = -\eta \sum_{j=1}^{J} \psi_j^m(t) \frac{\partial E_j^m}{\partial w^m} = -\eta \frac{\partial E^m}{\partial w^m} \tag{22}$$

$$u^j(t+1) = \sum_j \psi_j^m x_j \tag{23}$$

The processing mode, on the evaluative process, input from the robot evaluates the reference vector, and competitive process decides the BMU. The output is calculated by using BMU's MLP.

**Table 2.** Variable used Explanation of mnSOM Algorithm

| Symbol | Quantity |
|---|---|
| $v$ | Reference module vector |
| $x$ | Input vector |
| $y$ | Output vector |
| $j$ | Index expressing classes (j = 1, .... , J) |
| $m$ | Index of module |
| $Em$ | Distance between input vector and reference vector |
| $m^*$ | Best Matching Module (BMM) |
| $\phi$ | Neigubor Function |
| $d(a,b)$ | Euclidean distance between a and b |
| $\psi$ | Learning late |
| $\sigma$ | Neighbor radius |
| $\tau$ | Time constant |

## 4 Experiment

In order to evaluate the performances of color constancy algorithms, 15 sets of teaching data in various lighting conditions are prepared for learning. Table III shows the lighting conditions where the data are sampled and their luminance. The data are measured in fluorescent light (FL), fluorescent light sunshine (FL and SUN), white mercury lamp (WM), orange mercury lamp (OM) and under sunshine (SUN) and their luminance change from 15 to 22000 [lx]. The data sets in table IV are the test data for performance evaluation, not used in the learning process. The learning times for each algorithms are 30000.

**Table 3.** Environment of the Learning Data

| Light environment (Location) | Index (label) | Illuminance [lx] |
|---|---|---|
| FL (RoboCup room) | Robo015 | 15 |
| FL (RoboCup room) | Robo050 | 50 |
| FL (RoboCup room) | Robo110 | 110 |
| FL and SUN (RoboCup room) | Robo400 | 400 |
| FL and SUN (RoboCup room) | Robo473 | 473 |
| FL and SUN (Entrance) | Ent030 | 30 |
| FL and SUN (Entrance) | Ent200 | 200 |
| FL and SUN (Entrance) | Ent340 | 340 |
| SUN (Outside) | Out7600 | 7600 |
| SUN (Outside) | Ent22000 | 18640 |
| SUN (Gym) | SUN10 | 10 |
| WM and SUN (Gym) | Gym090 | 90 |
| OM and SUN (Gym) | Gym125 | 125 |
| WM, OM and SUN (Gym) | Gym133 | 133 |
| WM, OM and SUN (Gym) | Gym220 | 220 |

SUN: Sunshine, FL: fluorescent light,
WM: white mercury lamp, OM: orange mercury lamp)

**Table 4.** Environment of the Non Learning Data

| Light environment (Location) | Illuminance [lx] |
|---|---|
| FL (RoboCup room) | 310 |
| FL and SUN (RoboCup room) | 69 |
| SUN (RoboCup room) | 2 |
| FL and SUN (Entrance) | 337 |
| FL and SUN (Entrance) | 168 |
| SUN (Entrance) | 30 |
| SUN (Outside) | 58 |
| SUN (Outside) | 980 |

SUN: Sunshine, FL: fluorescent light,
WM: white mercury lamp, OM: orange mercury lamp

**Table 5.** Environment of the Non Learning Data

| | SOM | NG | mnSOM |
|---|---|---|---|
| Learning speed [sec] | 27.9 | 17.1 | 81.5 |
| Error of the learning data | 0.0007 | 0.0000 | 0.0367 |
| Execution speed [fps] | 16.3 | 17.1 | 8.7 |
| Recognize Rate of non learning data [%] | 83.3 | 85.4 | 85.4 |

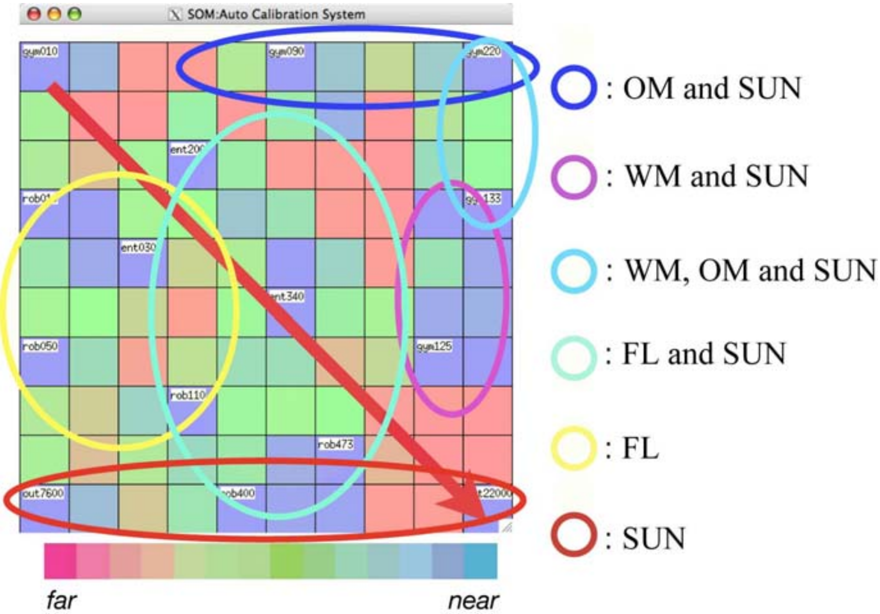


**Fig. 2.** Environment clustering result using SOM. Arrow means from dark area to light are. Circle means clusters depending on light conditions.

Table 5 shows the comparison of the learning results. With 30000 times learning, the errors of SOM and Neural Gas based algorithms converged to almost zero, but in mnSOM algorithm still error remains. The execution speed means how many frames can be processed online in the RoboCup robot. The SOM and NG based algorithms can calculate 2 times faster than mnSOM based algorithm. Recognition rate indicates how many objects can be recognized in non-learning data sets. Regards to the recognition rate, NG and mnSOM based algorithms show the best performance. In this problem, the NG based algorithm shows the best performance. mnSOM, an expansion of conventional SOM shows also good performance, however, this color constancy problem is not so complex as the mnSOM supposes.

Figure 2 shows an obtained feature map of environment clustering by learning various lighting conditions. Each lattice corresponds to a lighting environment, and each label means the Best Matching Unit (BMU) regarding learning environments (place and luminance [lx]). The color of lattice expresses the relative distance to it's neighborhood units, the red color means the unit is far from neighborhood lattice and blue does close. We can recognize the learning environmental data are clustered shown as the circles. The blue circle indicates the sunshine and white mercury lamp (SUN and WM), purple circle: sunshine and orange mercury lamp (SUN and OM), sky blue: sunshine and white mercury lamp and orange mercury lamp (SUN and OM and WH), green circle: sunshine and fluorescent light (SUN and FL), yellow circle: only fluorescent light (FL), and red circle only the sunshine (SUN). It can be seen that lighting conditions change gradually from dark to light (Arrow in Fig.2) and also color temperature.

## 5 Result

In this paper, we propose and develop the color recognition system of using the bio-inspired algorithm. Considering about mounting the autonomous mobile soccer robot, **NG** algorithm is the best of color detection algorithm.

In this algorithm, the algorithm detect the thresholds under the non-linear light environment, however, in the future work, the algorithms of bio-inspired can also adjust the camera parameters (Iris, white balance, gamma parameter and so on.)

## Acknowledgement

Many people and graduated students have helped us along the way. Their guidance, good humor, advice, and inspiration sustained us through the years of work. And, this work was supported by the 21st Century COE program, and Graint-in-Aid for Japan Society for the Promotion of Science (JSPS) Fellows.

## References

1. RoboCup Homepae, `http://www.robocup.org`
2. Matsubara, H., Asada, M., et al.: The RoboCup: The Robot World Cup Initiative. In: Proceedings of IJCAI 1995 (1995)
3. Takemura, Y., Nassiraei, A.A.F., et al.: HIBIKINO-MUSASHI. In: Proc. of RoboCup 2007, Atlanta, CD-ROM (2007)
4. Nassiraei, A.A.F., Takemura, Y., Sanada, A., et al.: Concept of Mechatronics Modular Design for an Autonomous Mobile Soccer Robot. In: CIRA 2007, Jacksonville, pp. 178–183 (2007)
5. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biol. Cybernetics 43, 59–69 (1982)
6. Heidemann, S.G., Ritter, H.: Parametrized SOMs for Object Recognition and Pose Estimation. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, pp. 902–907. Springer, Heidelberg (2002)
7. Barreto, G.A., Araujo, A.F.R., Ducker, C., Ritter, H.: A distribution of a simulator for robots and its application to position estimation by self-organization. Technical Report of IEICE 100(687), NC2000-147, 149–156

8. Ishii, K., Nishida, S., Ura, T.: A Self-Organizing Map Based navigation System for an Underwater Vehicle. In: Proc. of ICRA 2004, pp. 4466–4471 (2004)
9. Azeura, K., Godler, I.: Color Sampling with Omni-Directional Camera in Robot Soccer. In: Proc. of RSJ 2006, Okayama, 1B25.pdf (2006) (in Japanese)
10. Kohonen, T.: The self^organizaing map. In: Proceedings of the Instituite of Electrical and Electronics Engineers, vol. 78, pp. 1460–1480
11. Lo, Z.-P., Fujita, M., Bavarian, B.: Analysis of neighborhood interaction in Kohonen neural networks. In: 6th International Parallel Processing Symposium Proceedings, Los Alamitos, CA, pp. 247–249 (1991)
12. Lo, Z.-P., Yu, Y., Bavarian, B.: Analysis of the convergence properties of topology preserving neural networks. IEEE Transactions on Neural Networks 4, 207–220 (1993)
13. Erwin, E., Obermayer, K., Schulten, K.: I:Self-organizing maps: Stationary states, metastability and convergence rate. Biological Cbernetics 67, 47–55
14. Ritter, H., Martinets, T., Schulten, K.: Neural Computation and Self-Organizing Maps: An Introduction. Addison-Wesley, Reading
15. Obermayer, K., Ritter, H., Schulten, K.: Development and spatial structure of cortical feature maps: A model study. In: Advances in Neural Information Processing Systems, vol. 3, pp. 11–17. Morgan Kaufmann, San Mateo
16. Martinets, T.M., Berkovich, S.G., Schulten, K.J.: Neural Gas Network for Vector quanitization and its Application to Time-Series Prediction. IEEE Transactions on neural networks 4(4) (July 1993)
17. Tokunaga, T., Furukawa, T., Yasui, S.: Modular network SOM:Extention of SOM to the realm of function space. In: Proc. of Workshop on Self-Organizing Map 2003 (WSOM 2003), Kitakyushu, Japan, pp. 173–178 (2003)
18. Furukawa, T., Tokunata, T., Morishita, K., Yasui, S.: Modular network SOM (mnSOM): From vector space to function space. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3696, pp. 391–396. Springer, Heidelberg (2005)

# Surprise-Driven Exploration with Rao-Blackwellized Particle Filters for Efficiently Constructing Occupancy Grid Maps

Youbo Cai and Masumi Ishikawa

Department of Brain Science and Engineering, Kyusyu Institute of Technology
Kitakyusyu 808-0196, Japan
cai-youbo@edu.brain.kyutech.ac.jp,
ishikawa@brain.kyutech.ac.jp

**Abstract.** Proposed is a novel algorithm for surprise-driven exploration with Rao-Blackwellized Particle Filters (RBPF) for simultaneous localization, mapping, and exploration. ”Exploration” is expected to find the optimal path for efficient simultaneous localization and mapping (SLAM). We propose to adopt the concept of *surprise* defined by Kullback-Leibler (KL) divergence between posterior belief and prior belief. During exploration, it evaluates each path by trading off its cost against the degree of surprise for the corresponding expected map in RBPF. We also propose to automatically generate candidate paths for evaluation instead of providing them from the outside. Simulation experiments demonstrate the effectiveness of the proposed algorithm. Compared with the previous studies that use information gain-driven exploration, the proposed method shows superior performance in selecting path closing, which drastically reduces uncertainty of robot poses and accordingly improves the accuracy of the resulting map.

**Keywords:** surprise, SLAM, Rao-Blackwell, particle filter, exploration.

## 1 Introduction

To develop truly autonomous mobile robots capable of behaving in unknown environments, localization and mapping need to be solved simultaneously. This is because a precise map can only be acquired from a well-localized robot, but the accuracy of localization relies on the precision of the map. This is the so-called simultaneous localization and mapping (SLAM) problem [3]. One of popular solutions to SLAM is the Rao-Blackwellized Particle Filters (RBPF or FastSLAM), the efficiency of which was significantly improved by Montemerlo et al. [5] and Grisetti et al. [6][7].

SLAM, however, does not provide information on where to go next for obtaining sensory signals needed for efficient SLAM. The quality of a resulting map heavily depends on a path during exploration [8][9]. Hence, an integrated approach to simultaneous localization, mapping, and exploration is needed. Yamauchi proposed to heuristically steer a robot to continuously explore a new

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 420–429, 2009.

terrain to reduce the uncertainty in the map [11]. Stachniss et al. proposed to actively close loops to reduce uncertainty in the robot path [8]. Other approaches that take into account uncertainty in both the map and a robot path evaluate a path by trading off its cost against the amount of *information gain* calculated from pose-map joint entropy[9] or expected map entropy [10] with RBPF.

We seek an alternative brain-inspired approach instead of an engineering-oriented approach using *information gain*. Vital to survival of animals is their ability to notice the change of the environment and to adjust to it by adaptively deciding suitable actions [1]. Itti et al. proposed *surprise* defined by Kullback-Leibler (KL) divergence between posterior belief and prior belief of an agent about the world [2]. They also provided experimental evidences that the proposed *surprise* well characterizes what attracts human attention in video stream [2]. Our observation is that we continuously predict future events (prior belief) and verify them by incoming sensory signals (posterior belief). Based on the observation, we believe that *surprise* is a promising alternative brain-inspired approach which is expected to have good performance in evaluating a path during exploration than *information gain*. It is to be noted that a brain-inspired approach is different from a brain-mimicking approach. The latter has the corresponding mechanism in the brain, hence is not applicable when the brain mechanism is unknown. In contrast to this, the former is applicable without the corresponding brain mechanism and with an adequate hypothesis on the mechanism.

Therefore, we propose to use *surprise* for evaluating a path by trading off its cost against the degree of *surprise* for the expected map, and to compare the performance of surprise-driven exploration and the conventional information gain-driven exploration. We also propose to automatically generate candidate paths by a depth-bounded search algorithm, assuming the availability of a topological map. In the conventional studies, paths were given from the outside instead of being generated automatically. Simulation experiments are done to demonstrate superiority of the proposed surprise-driven exploration over the information gain-driven exploration in selecting path closing, which drastically reduces uncertainty of robot poses and accordingly improves the accuracy of the resulting map.

The rest of the paper is organized as follows. Section 2 presents our surprise-driven exploration algorithm. Section 3 demonstrates experimental results and compares the performance between surprise-driven exploration and information gain-driven exploration. Section 4 concludes the paper.

## 2 Surprise-Driven Exploration Algorithm

We first formulate *surprise* for the expected map in RBPF, present an algorithm for constructing a topological map as a prerequisite for automatic path generation, and finally propose an algorithm for surprise-driven exploration with RBPF.

### 2.1 Formulation of Surprise for the Expected Map

A key idea in SLAM is to estimate a joint posterior $p(x_{1:t}, m|z_{1:t}, u_{1:t-1})$ over the robot path $x_{1:t} = x_1, x_2, \cdots, x_t$ along with the map $m$, given its observations $z_{1:t} = z_1, z_2, \cdots, z_t$ and odometry measurements $u_{1:t-1} = u_1, u_2, \cdots, u_{t-1}$. According to Murphy [4], the joint posterior can be factorized as,

$$p(x_{1:t}, m|z_{1:t}, u_{1:t-1}) = p(x_{1:t}|z_{1:t}, u_{1:t-1})p(m|x_{1:t}, z_{1:t}) \tag{1}$$

Eq.(1) can be obtained efficiently, because it allows to first estimate only the path of the robot and then to compute the map based on that path. To estimate the posterior $p(x_{1:t}|z_{1:t}, u_{1:t-1})$ , RBPF uses particle filters (PF), in which each particle carries its own map. Each map is computed by the ray tracing based on the observations $z_{1:t}$ and the path $x_{1:t}$ represented by the corresponding particle. The weight of each particle is computed as proportional to the likelihood $p(z_t|m, x_t)$ of the most recent observation $z_t$, given the pose $x_t$ represented by the corresponding particle and the associated map $m$.

The expected map obtained by marginalizing out the robot paths is [10],

$$p(\bar{m}_i|d_t) \approx \sum_{k=1}^{M} \omega_t^k p(m_i^k|x_t^k, d_t) \tag{2}$$

where $M$ is the number of particles in RBPF, $t$ is a time step, $k$ is a particle number, $x_t$ is the pose, $\omega_t^k$ is its importance weight, $m_i^k$ is the occupancy probability of the $i$th grid cell, and $d_t$ comprises the observation $z_t$ and the odometry measurement u_t-1.

Because occupancy probability of each grid cell in the expected map is represented by a binary random variable with Bernoulli distribution, the degree of surprise for the $i$th grid cell is given by [2],

$$S(\bar{m}_i) = q_i \log \frac{q_i}{p_i} + (1 - q_i) \log \frac{1 - q_i}{1 - p_i} \tag{3}$$

where $p_i$ and $q_i$ stand for prior and posterior belief of the $i$th grid cell, respectively. Assuming that these random variables are mutually statistically independent as in RBPF, the degree of surprise for the whole expected map is given by,

$$S(\bar{m}) = \sum S(\bar{m}_i) \tag{4}$$

Similarly, information gain for the $i$th grid cell and that for the whole expected map are,

$$I(\bar{m}_i) = q_i \log q_i + (1 - q_i) \log(1 - q_i) - p_i \log p_i - (1 - p_i) \log(1 - p_i) \tag{5}$$

$$I(\bar{m}) = \sum I(\bar{m}_i) \tag{6}$$

### 2.2 Algorithm for Constructing a Topological Map

Surprise-driven exploration with RBPF in 2.3 assumes the availability of a topological map. We previously constructed a topological map based on infra-red sensors [12]. We think it is also possible to construct a topological map based on omni-view images.

Suppose a topological map initially has only one node corresponding to the starting position of the robot as in Fig. 1(a). The topological map, represented by a graph $G$, is constructed by the following procedure.

- Add a new node to $G$, when the robot has moved 1 m from its initial position.
- Add a new node to $G$, when the robot detects a corner or a crossroads.
- Add a new node to $G$, when the distance between the robot and any visible node in $G$ exceeds 6 m.
- Add a new edge between the current node and the previous node to $G$, when a new node is added.
- Add new edges between the current node and visible nodes within 6 m, when the robot reaches a node.

In the very beginning stage, a set of sensory information obtained is not enough. Therefore, an extra node is created where the robot moves 1 m away from its initial position. In Fig. 1(a), starting from the node *s* and passing through the nodes *1,2,...,10*, the graph with 12 nodes and 12 edges is constructed.

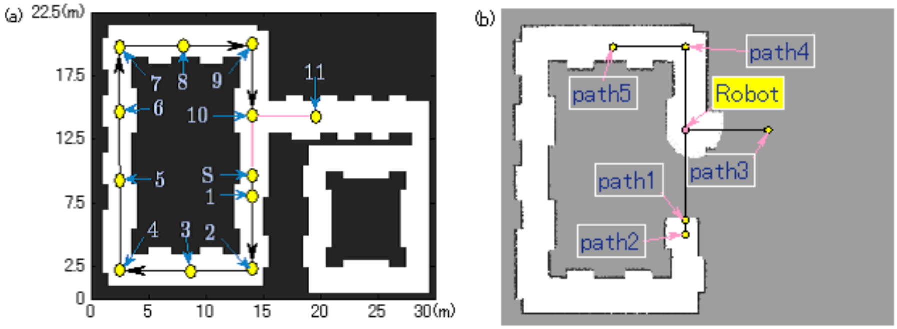


**Fig. 1.** (a)The robotic field and the resulting topological map. The black stands for occupied grids, and the white stands for free grids. Starting from the node *s* and passing through the nodes *1,2,...,10*, the robot constructs a graph with 12 nodes and 12 edges. (b) The resulting occupancy grid map and the candidate paths when the robot arrives at the node 10.

### 2.3 Algorithm for Surprise-Driven Exploration

A key idea of the proposed algorithm is to evaluate each path by trading off its cost against the degree of surprise for the expected map in RBPF. Another idea is to automatically generate candidate paths from a topological map being constructed.

Let a graph be denoted by the matrix $G = \{< g_{ij}, s_{ij}, l_{ij} > | i, j = 1, ..., N\}$, where $N$ is the number of nodes in $G$; $g_{ij} = 1$ if the node $i$ is visible from the node $j$, otherwise $g_{ij} = 0$; $s_{ij}$ denotes the degree of surprise for the expected map when the robot moves from the node $i$ to the node $j$; $l_{ij}$ denotes the path length between the node $i$ and the node $j$. Suppose a path has $K$ nodes and is represented by a linear list $lst = (Node_1, Node_2, ..., Node_K)$. Assuming the cost of a path is proportional to its length, the utility function is given by,

$$f_{lst} = Surprise(lst) - Cost(lst) = \sum_{<i,j> \in lst} s_{ij} - \alpha \sum_{<i,j> \in lst} l_{ij} \tag{7}$$

$$lst^* = arg \max_{lst} f_{lst} \tag{8}$$

where $\alpha$ is a weighting factor and $lst^*$ corresponds to the selected path. We determine $\alpha$ experimentally.

**Table 1.** Algorithm for Surprise-driven Exploration

```
**Surprise-drivenExploration** $(G, n_0, \Lambda_{n_0})$
    Initialize the path set $\Omega = \emptyset$
    Initialize the current path $list$ with the current node $n_0$
    Set $depth$= 1
    Call $[G, \Omega]$=Depth-boundedSearch($G, n_0, \Lambda_{n_0}, list, depth, \Omega$)
    For each path $lst \in \Omega$
        Compute utility function value $f_{lst}$ by Eq.(7)
    End for
    Return $arg \max_{lst} f_{lst}$
```

**Table 2.** Algorithm for Depth-bounded Search

```
**Depth-boundedSearch**$(G, i, \Lambda_i, list, depth, \Omega)$
    For $j = 1 : N$
        If($g_{ij} = 1$ AND $j$ does not exist in $list$)
            Add the node $j$ to the last node of $list$
            Call $[s_{ij}, \Lambda_j]$=$CalculateSurpriseBetweenTwoNodes(i, \Lambda_i, j)$
            Save $s_{ij}$ to $G$
            Add the current path $list$ to $\Omega$
            If($depth < MaxDepth$)
                $depth = depth + 1$
                Call $[G, \Omega]$=Depth-boundedSearch($G, j, \Lambda_j, list, depth, \Omega$)
                $depth = depth - 1$
            End if
            Delete the last node from current path $list$
        End if
    End for
    Return $[G, \Omega]$
```

**Table 3.** Algorithm for Calculating Surprise Between Two Adjacent Nodes

```
 1:CalculateSurpriseBetweenTwoNodes($i, \Lambda_i, j$)
 2:    Set $s_{ij} = 0, \Lambda_j = \emptyset$
 3:    For $h = 1 : H$
 4:        Randomly select a member $[< x_t^k, m_t^k >, k_t^*]$ from $\Lambda_i$
 5:        Generate actions $u_{t:T}$ which make the robot move from $i$ to $j$
 6:        For $\tau = t : T$
 7:            Draw $x' \sim p(x'|x_\tau^{k_\tau^*}, m_\tau^{k_\tau^*}, u_\tau)$
 8:            Draw $z_\tau \sim p(z|x', m_\tau^{k_\tau^*})$
 9:            Call $[< x_{\tau+1}^k, m_{\tau+1}^k >, k_{\tau+1}^*]$=RBPF($< x_\tau^k, m_\tau^k >, z_\tau, u_\tau$)
10:        End for
11:        Add $[< x_T^k, m_T^k >, k_T^*]$ to $\Lambda_j$
12:        Calculate the expected map by Eq.(2)
13:        Calculate surprise $s_h$ for the whole expected map by Eq.(3-4)
14:        $s_{ij} = s_{ij} + s_h/H$
15:    End for
16:    Return $[s_{ij}, \Lambda_j]$
```

Table 1 gives the proposed algorithm for surprise-driven exploration. The inputs to the algorithm are the graph $G$, the current node $n_0$, and a data set $\Lambda_{n_0} = \left\{< x_t^k, m_t^k >, k_t^*\right\}$, which comprises the particles and the best particle number at time $t$. The output is the optimal path with the maximal utility. A depth-bounded search algorithm in Table 2 generates all paths starting from the current node by searching the graph G recursively with a limited depth (MaxDepth). Table 3 shows the algorithm for calculating the degree of surprise between two adjacent nodes using RBPF as a simulator (Line 9).

## 3 Experimental Results

Simulation experiments are carried out to demonstrate the effectiveness of the proposed method and to compare its performance with that by information gain-driven exploration from the viewpoints of map errors and position errors.

Fig. 1(a) also illustrates the robotic field. It has 225x300 grid cells with each grid being 0.1 m by 0.1 m. The robot is circular with the radius of 0.2 m and is equipped with a scanning laser ranger finder. Its view angle is 180° with 1° angular resolution. Its sensing range is 2.0 m with error Std. of 4 cm. A control action is a combination of rotation and forward movement. We assume that a control action is corrupted by a Gaussian noise with angular Std. of 0.1° and velocity Std. of 0.5m/s. Fifty particles are employed to represent prior belief and posterior belief. The number of particles required depends on the noise level; the larger the noise level is, the larger the number of particles becomes. The initial occupancy probability of each grid cell is assumed to be 0.5 (unknown). The weighting factor $\alpha$ is 80 for both surprise-driven and information gain-driven utilities.

### 3.1 Path Generation

Fig.1(a) illustrates that the topological map with 12 nodes and 12 edges stating from the initial node $s$ and ending at the node 10. Fig.1(b) shows that five candidate paths are automatically generated for evaluation by searching the topological map with the maximum depth of 2.

Fig.2(a) and Fig.2(b) show that surprise-driven exploration selects the path 2, while information gain-driven exploration selects the path 3. Fig.2(c) and Fig.2(d) demonstrate that surprise-driven exploration selects path closing at time step 115, while information gain-driven exploration selects path closing at time step 141. For time steps 1 through 114, surprise-driven exploration and information gain-driven exploration give the same path, because there exists no branch point on the path in both Fig.2(c) and Fig.2(d).

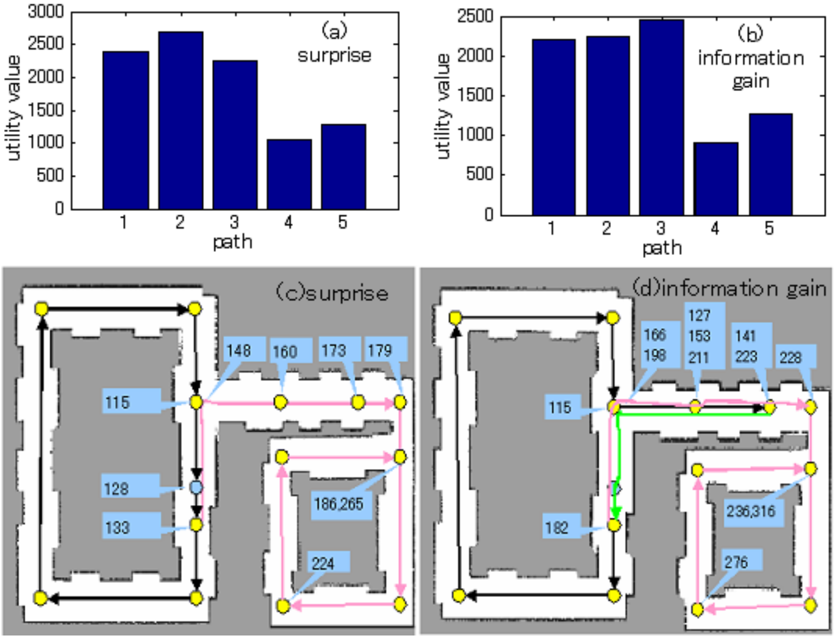


**Fig. 2.** (a)Utility values for candidate paths by surprise-driven exploration. (b)Utility values for candidate paths by information gain-driven exploration. (c)The resulting path and time steps by surprise-driven exploration. (d)The resulting path and time steps by information gain-driven exploration.

### 3.2 Performance Comparison

Fig. 3(a) and Fig. 3(b) illustrate map MSEs and position errors, respectively, for surprise-driven exploration with RBPF and information gain-driven exploration with RBPF. They demonstrate that surprise-driven exploration with RBPF is

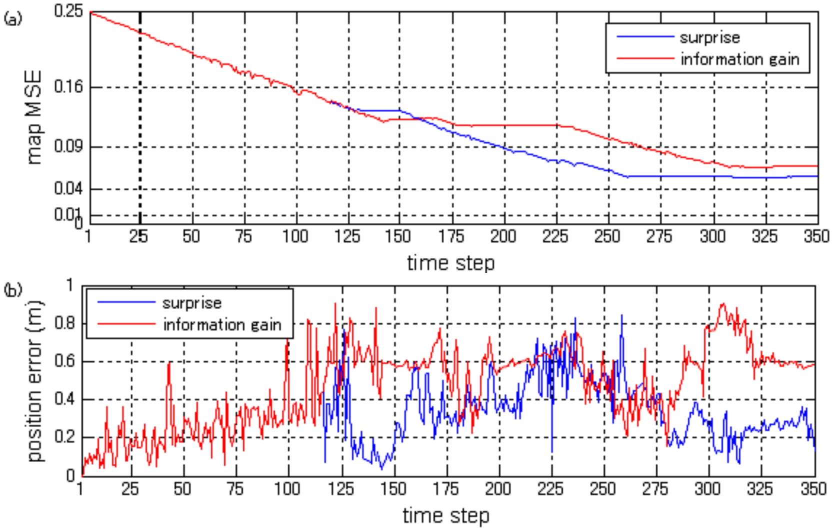


**Fig. 3.** (a)Comparison of map errors. (b)Comparison of position errors. From time step 1 to 114, both methods give the same path, hence produce the same result for two methods under the condition that the same randomness is adopted by two methods.

superior to information gain-driven exploration with RBPF in accuracy of the resulting map and localization. This is mainly because surprise-driven exploration selects path closing earlier than information gain-driven exploration, which drastically reduces its path uncertainty and accordingly improves the quality of the resulting map.

To consider relative position error, we adopt the length between two adjacent nodes as a reference, which is typically 6m. At time step 150, the relative position error by surprise-driven exploration is 3.3%, and that by information gain-driven exploration is 10.0%.

## 4 Conclusions and Discussions

The present paper proposed a surprise-driven exploration method with Rao-Blackwellized particle filters and compared the performance between the proposed method and the information gain-driven method. Experimental results demonstrated that the proposed algorithm can automatically generate candidate paths for exploration and performs better than the conventional information gain-driven method in accuracy of the resulting map and localization. This is because the surprise-driven exploration has stronger tendency to select actions to close a loop, which drastically reduces the uncertainty of the robot pose and accordingly improves the quality of the resulting map.

The values of parameters used in the paper, i.e., noise levels for a laser range finder, velocity control, and angular control, are considered to be typical. The resulting map MSEs and position errors, however, depend on the noise levels. Generally speaking, two methods provide the same result; at high noise levels both methods select path closing and at small noise levels both methods select exploration in a new terrain. At intermediate noise levels, two methods provide different results; surprise-driven exploration selects path closing, and information gain-driven exploration selects exploration in a new terrain followed by path closing later. The detailed analysis of the dependency of the performance on these noise levels will be presented in the near future.

Examination of the robustness of the proposed method and its application to different robotic fields are left for future studies. How to detect a corner or a crossroads using omni-view images in constructing a topological map is also left for further study.

**Acknowledgments.** This research was partially supported by Grant-in-Aid for Scientific Research(C) (21500218) from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

## References

1. Ranganath, C., Rainer, G.: Nat. Rev. Neurosci. 4, 193–202 (2003)
2. Itti, L., Baldi, P.: Bayesian Surprise Attracts Human Attention. In: Advances in Neural Information Processing Systems (NIPS 2005), vol. 19, pp. 547–554. MIT Press, Cambridge (2006)
3. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cambridge (2005)
4. Murphy, K.: Bayesian Map Learning in Dynamic Environments. In: Advances in Neural Information Processing Systems (NIPS), vol. 12, pp. 1015–1021 (1999)
5. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping That Provably Converges. In: Proc. of the Int. Conf. on Artificial Intelligence (IJCAI), pp. 1151–1156 (2003)
6. Grisetti, G., Stachniss, C., Burgard, W.: Improving Grid-based Slam with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), pp. 2443–2448 (2005)
7. Grisetti, G., Tipaldi, G., Stachniss, C., Burgarda, W., Nardib, D.: Fast and Accurate SLAM with Rao-Blackwellized Particle Filters. Robotics and Autonomous Systems 55(1), 30–38 (2007)
8. Stachniss, C., Hahnel, D., Burgard, W.: Exploration with Active Loop-closing For FastSLAM. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), vol. 2, pp. 1505–1510 (2004)

9. Stachniss, C., Grisetti, G., Burgard, W.: Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. In: Proceedings of Robotics: Science and Systems (RSS), pp. 65–72 (2005)
10. Blanco, J.L., Fernandez-Madrigal, J.A., Gonzalez, J.: A Novel Measure of Uncertainty for Mobile Robot SLAM with Rao-Blackwellized Particle Filters. The International Journal of Robotics Research 27(1), 73–89 (2008)
11. Yamauchi, B.: Frontier-based Exploration Using Multiple Robots. In: Proceedings of the Second International Conference on Autonomous Agents, pp. 47–53. ACM Press, New York (1998)
12. Aziz, M., Ishikawa, M.: Formation of Graph-based Maps for Mobile Robots Using Hidden Markov Models. In: IEEE World Congress on Computational Intelligence (WCCI 2008), Hong Kong, pp. 3098–3104 (2008)

# Retrieving Emotion from Motion Analysis: In a Real Time Parallel Framework for Robots

Tino Lourens and Emilia Barakova

Eindhoven University of Technology
P.O. Box 513, Eindhoven, The Netherlands
t.lourens@tue.nl, e.i.barakova@tue.nl

**Abstract.** This paper presents a parallel real time framework for emotion extraction from video fragments of human movements. Its framework is used for tracking of a waving hand by evaluation of moving skin-colored objects. The tracking analysis demonstrates that acceleration and frequency characteristics of the traced objects are relevant for classification of the emotional expressiveness of human movements. The solution is part of a larger project on interaction between a human and a humanoid robot with the aim of training social behavioral skills to autistic children with robots acting in a natural environment.

## 1 Introduction

Sociable humanoid robots pose a dramatic and intriguing shift in the way one thinks about control of autonomous robots, and are the first generation of robots where a substantial human-robot interaction is expected. The introduction of mobile robots that have to demonstrate a certain degree of constitutive autonomy yield different requirements than industrial robots that have been preprogrammed to work in a fully controlled environment. Sociable robots need to have an even higher level of autonomy, dealing not only with its perceptual, and behavioral aspects, but also with its interactive aspects, as present for instance in the emotion system [5,2]. In addition also a mechanism to cooperate with uncertainty and survival is needed to guarantee a degree of autonomy. In many ways such a system resembles aspects of brain like functional behavior, its evident that such a robot should be able to process information in real time in a highly parallel way. We have adopted the approach of functional brain modeling [16] and use graphical software environment TiViPE [12] to realize and integrate these functional models, in a similar way as in earlier work [19,15,13,14]. Real time parallelism on a PC has been strongly facilitated by recent developments of graphical processing units (GPUs), not only have these GPUs become fast[1] but they also can be used as general processing units [8].

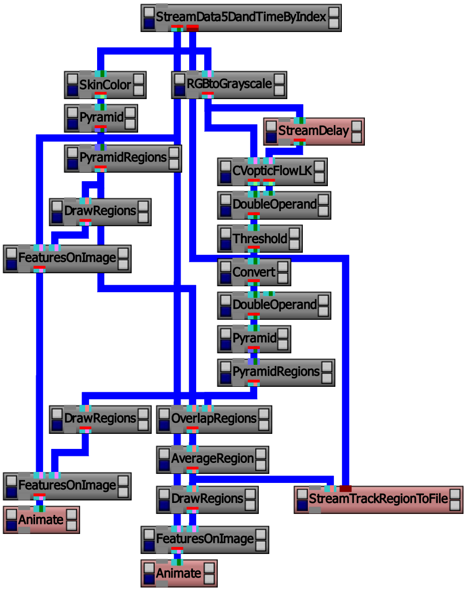
We are interested in scenarios involving multiple simultaneous actions performed by different body parts of a human or a robot. We assume realistic

[1] A single GPU card is able to process more than one tera ($10^{12}$) floating point operations per second (TFLOPS).

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 430–438, 2009.

imitation scenarios, i.e., scenarios where a human freely behaves and a robot tracks its actions with the intend to act upon the meaningful ones, for instance by imitation [3] or by encoding episodes [1]. In this paper the focus is on hand waving with the aim of detecting different emotional states that can be used either to imitate or to influence the emotional state.

The paper is organized as follows: Section 2 describes the experimental setup and gives the implementation of marking a moving hand in an image using skin color and motion characteristics. For the sequence of images such region is marked to construct a stream that is used to analyze hand waving behavior. Section 3 provides insight how these data streams are used to extract social behavior for interaction with a robot. The paper finishes with a discussion and future research.



1. acquiring data from a camera or reading a stored image sequence
2. binarizing an image by marking a pixel either as skin color or other color and in parallel binarizing an image by marking pixels either as observed motion or as static element
3. marking skin and motion regions by a pyramid decomposition
4. selection of these regions that are both skin and motion region
5. averaging skin-in-motion regions to a single region
6. tracking an averaged skin-in-motion region
7. visualization of regions in an image
8. visualization of a waving hand
9. classification of waving profiles

**Fig. 1.** TiViPE (www.tivipe.com) implementation of handwaving. The icons from top to bottom at the left-side process skin areas, while motion sensitivity is processed by the functional blocks at the right-side.

## 2 Experimental Setup

We have been conducting hand waving experiments within scenarios where different emotions has been enacted. Figure 3 depicts four (happy, angry, sad, and polite) different emotional waving patterns. A camera records images that are processed using a combination of skin color and motion detection, with the aim of tracking a single area. This area is associated with the waving movement.

A device or robot should be able to extract a simple motion pattern and interpret the intend or the emotion of this movement behavior, imitate this movement pattern or eventually adjust its own behavior. The aim of the overall project is to teach or to influence behavior of the human in order to improve his or her social skills.

The implementation of detection and tracking a waving hand is given in Figure 1. The theoretical background and the computational details behind the implemented signal processing is described in more detail in the following subsections.

### 2.1 Skin Color Detection

An image is defined as a two-dimensional matrix where a pixel at position $(x, y)$ has an intensity $I_c(x, y) = (r, g, b)$, where $r$, $g$, and $b \in [0, \ldots, 255]$ are the red, green, and blue component. Segmentation using skin color can be made independent of differences in race when processing image pixels in Y-Cr-Cb color space [6]. The following $(r, g, b)$ to $(Y, Cr, Cb)$ conversion is used

$$Y = 0.2989r + 0.5866g + 0.1145b, \quad Cr = 0.7132(r - Y), \quad Cb = 0.5647(b - Y),$$

where threshold values $77 < Cb < 127$ and $133 < Cr < 173$, see [6] yield good results for classifying pixels belonging to the class of skin tones.

In our experiments we also excluded the “white area”. Formally an element belongs to the “white area” if it satisfies the following:

$$\frac{|r-g|}{m} < 0.1 \wedge \frac{|r-b|}{m} < 0.1 \wedge \frac{|g-b|}{m} < 0.1, \tag{1}$$

where $m = \min(r, g, b)$, $r > 0.3$, $g > 0.3$, and $b > 0.3$. Its implementation given by ‘the “SkinColor” icon in Figure 1 yields a binary image.

The functional concept as described above contains similarities with how the brain processes visual data, especially in the way the primary visual cortex area V4 provides a substantial role in processing color [20].

### 2.2 Motion Detection

A pixel at location $(x, y)$ with intensity $I(x, y) = (r + g + b)/3$ will have moved by $(\delta x, \delta y)$ over a time span $\delta t$, hence the following image constraint equation can be given:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t). \tag{2}$$

In this method the following needs to be solved:

$$I_x V_x + I_y V_Y = -I_t, \tag{3}$$

where $(V_X, V_y)$ denotes the flow, $I_x$ and $I_y$ the derivatives in horizontal and vertical direction, respectively. The Lucas-Kanade operator [17] is used, it is a two-frame differential method for optical flow estimation where the derivatives are obtained by using a Sobel filter kernel [18]. Instead of using Sobel kernels the more biologically plausible Gabor kernels can be used as well [7,11]. Receptive fields of a cat's primary visual cortex area V1 and V2, that show striking similarities with these Gabor kernels, have been found in the 1950's by neuroscientists and Nobel laureates Hubel and Wiesel [9,10]. It is plausible that a similar activity flow map might be expected in the middle temporal are MT, also known as primary visual cortex area V5 [20].

From $(V_x, V_y)$ the L-2 norm (top right "DoubleOperand" icon of Figure 1) is taken and thresholded at 5 to obtain a binary "motion classified" image.

### 2.3 Rectangular Region Marking

The next stage is marking a cluster of "skin tone classified" or "motion classified" pixels by a rectangular window. This is performed by decomposing the image into a pyramid, where every pixel in the next level of the pyramid is computed as follows:

$$I_{i+1}(x,y) = (I_i(2x,2y) + I_i(2x+1,2y) + I_i(2x,2y+1) + I_i(2x+1,2y+1))\,/4, \tag{4}$$

where $(x, y)$ is the position in image $I_i$, $i$ denotes the level in the pyramid. Base level 0 contains the original image $I_0$. The construction of a pyramid using (4) provides a strongly reduced search space, since if in level $i+1$ a pixel $I_{i+1}(x, y)$ is found to belong to the desired region then in level $i$ of the pyramid a cluster of

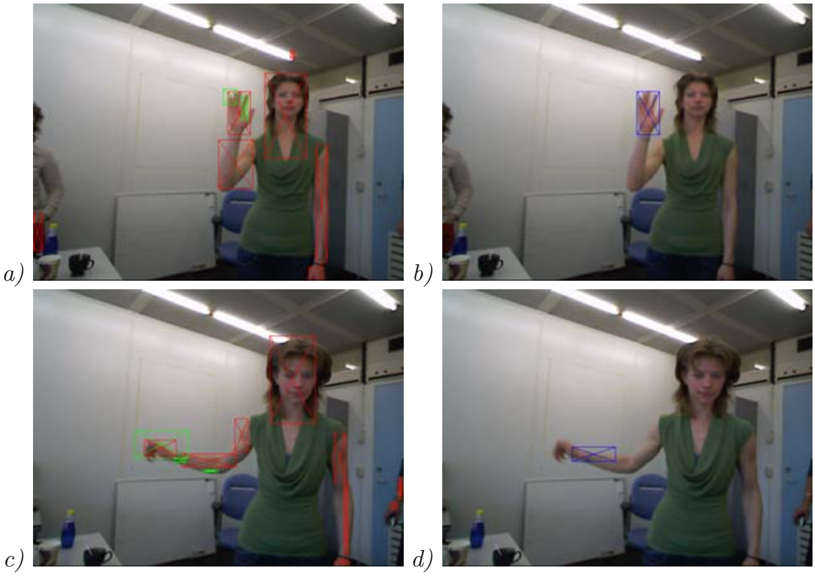


**Fig. 2.** Marked regions of interest. Red and green areas denote skin and motion area, respectively. A blue area is the combined moving skin area that has been averaged over all skin areas that are moving.

2x2 pixels ($I_i(2x, 2y)$, $I_i(2x+1, 2y)$, $I_i(2x, 2y+1)$, and $I_i(2x+1, 2y+1)$) belong to the same region.

The search for regions of interest starts at the highest level, and decreases until an a-priori known minimum level has been reached. It is therefore possible that no regions of interest are found. Taking into consideration that if a pixel is marked as "skin tone" or "motion" it has value 1, and 0 otherwise. We define a pixel to belong to a unique region $j$ if it satisfies the following:

$$R_i^j(x, x+1, y, y+1) = I_i(x, y) \equiv 1. \tag{5}$$

Regions $R_i^j$ in their initial setting are bound by a single pixel $I_i(x, y)$, and a region growing algorithm is applied to determine the proper size of the rectangular region. Lets assume that the initial size of the rectangle is $R_i^j(x_l, x_r, y_u, y_d)$ and that the possible growing areas are left ($R_i^{j^l} = R_i^j(x_l - 1, x_l, y_u, y_d)$), right ($R_i^{j^r} = R_i^j(x_r, x_r + 1, y_u, y_d)$), above ($R_i^{j^u} = R_i^j(x_l, x_r, y_u - 1, y_u)$, and below ($R_i^{j^d} = R_i^j(x_l, x_r, y_d, y_d + 1)$ this region. The average value of all four growing areas is taken, where the maximum value determines the direction of growing. The following procedure

$$A_i^{j^x} = \text{avg}\left(R_i^{j^x}\right), x \in \{l, r, u, d\},\ M_i^{j^x} = \max_x \left(A_i^{j^x}\right),\ R_i^j = R_i^j \cup R_i^{j^x}, \text{if } M_i^{j^x} \geq T_{rg}$$

is repeated until $M_i^{j^x} < T_{rg}$. From experiments $T_{rg} = 0.67$ provides a rectangle that corresponds roughly to a skin area in the original image and 0.5 gives a sufficiently large motion area, see also Figure 2.

The method described above is able to find all uniform skin color and motion regions in an image in real time.

Formally such a feature $f$ can be described by its region, type, and time: $f(xl, xr, yu, yd, regiontype, t)$. This $f$ in turn could be further processed by other visual areas or passed on to both STS and PFC.

### 2.4 Tracking

Two examples of the waving experiment using color images of 640x480 pixels at a speed of 29 frames per second are provided in Figure 2. Creating a single region in the image rather than multiple regions of interest is accomplished in order to unambiguously track an object of interest. These tracked objects are stored as file, see also icon "StreamTrackRegionToFile" of Figure 1, and processed further.

Fifteen recordings of 20 seconds have been made where a performer was asked to demonstrate happiness, anger, sadness, or politeness. In each plot the acceleration profile has been obtained by taking the second derivative of the central point of the tracked object. Examples for all four different emotional states are depicted in Figure 3. From this figure the following can be observed:

1. *happy* waving provides a regular waving pattern with a relatively high frequency.
2. *anger* demonstrates bursts with tremendous acceleration

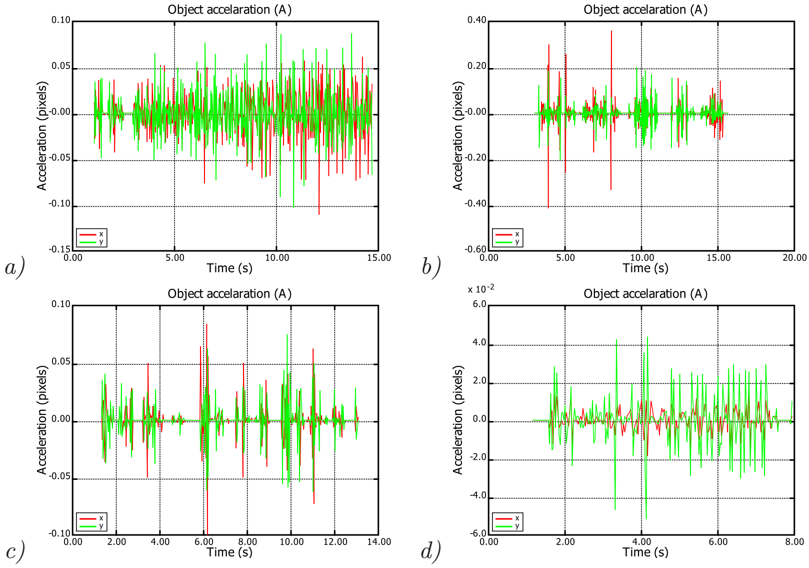


**Fig. 3.** Waving patterns. First row shows acceleration profiles for happiness, and anger. The second row provides sadness, and politeness profiles.

3. *sadness* demonstrates a profile of low acceleration, its frequency is relatively low and appears to have a lower frequency compared to the other three emotions.
4. *politeness* that demonstrates a queen type of waving profile is a regular pattern with a high frequency that is obtained by using minimal energy.

In an average acceleration-frequency plot of the recorded movements four distinctive clusters are formed (Figure 4). In one of the image sequences the actor was instructed to perform polite waving, but in the sequence she seemed to be happy, indicating that there might be a smooth boundary between these emotional states. The average energy in one of the five bursts in Figure 3b shows an average acceleration score of more than 0.07 and gives an indication of the upper bound of used energy by performing these emotions by the actor.

## 3 Behavioral Primitives

Understanding motion from waving patterns requires a mechanism that is able to learn to interpret and classify these sequences, and ideally able to extract the observations provided in Section 2.4. In a complementary study we are attempting to classify motion by so-called Laban primitives [2]. Using these primitives we classify the intend and the mental state of the person that perform movement behavioral patterns.

The current method is developed to enable a robot to interact with a human in realistic scenarios. If a robot is able to track in parallel regions of interest, a

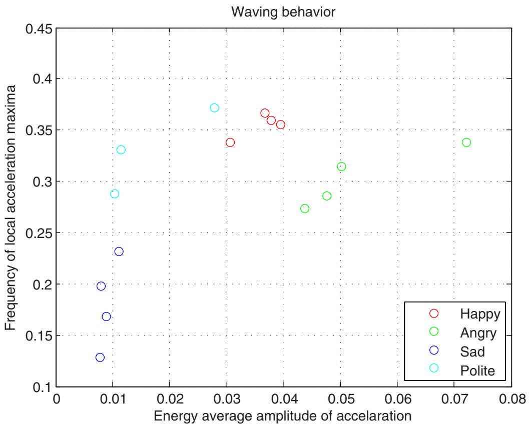


**Fig. 4.** Distinct emotion profiles are revealed by average frequency and acceleration

considerable number of interacting scenarios are possible even without interpreting of the meaning of an object. Moreover, using an earlier developed technique [4] the robot recognizes and learns repeating patterns of behavior, which it considers important, and discards occasional movements which most often are not important. For instance, if during waving of the hand a head movement takes place because at this time somebody enters the room, this movement will be ignored. However, if a person that interacts with the robot performs repeatedly a movement with his/her head while waving, this will be learned and eventually included in the imitation behavior of the robot.

## 4 Discussion and Future Work

In this paper we have shown that a device or robot is able to detect a waving hand in real time by using a combination of skin tone and motion. Tracking a moving hand provides clear insight about the emotional state of a person when displayed in frequency-amplitude domain. The four emotions performed by one person are clearly clustered and segregated in this domain making the current approach suitable for emotion recognition. In a complementary study these regions are transfered into behavioral primitives. These primitives are used by the robot to socially interact with a human.

It is obvious that we have barely touched the surface of the overall research that we would like to conduct. Even a simple experiment like hand waving elicits a number of questions:

- Could any type of waving be predicted?
- How to determine the differences between people?
- How to respond to a waving pattern?
- Does it lead to adaptive or predictive behavior?

- How does the design of simple interaction behavior look like?
- How to design imitation, such that it appears natural and distinctive on a humanoid robot?

An important design aspect of a humanoid robot will be how closely human movement can be emulated on it. The emulation will be restricted by the understanding the physical limitations of the robots, and the mechanism that cause the movement behavior. Next, a basic set of motion primitives needs to be derived that independently of the physical embodiment emulates body movements that are interpreted by humans as emotional and social.

## References

1. Barakova, E.I., Lourens, T.: Efficient episode encoding for spatial navigation. International Journal of Systems Science 36(14), 877–885 (2005)
2. Barakova, E.I., Lourens, T.: Analyzing and modeling emotional movements: a framework for interactive games with robots. Personal and Ubiquitous Computing (2009) (in press)
3. Barakova, E.I., Lourens, T.: Mirror neuron framework yields representations for robot interaction. Neurocomputing 72(4-6), 895–900 (2009)
4. Barakova, E.I., Vanderelst, D.: From spreading of behavior to dyadic interaction -a robot learns what to imitate. International Journal of Intelligent Systems (2009) (in press)
5. Breazeal, C.: Emotion and sociable humanoid robots. International Journal of Human-Computer Studies 59, 119–155 (2003)
6. Chai, D., Ngan, K.N.: Face segmentation using skin-color map in videophone applications. IEEE Transactions on Circuits and Systems for Video Technology 9(4), 551–564 (1999)
7. Daugman, J.G.: Complete discrete 2-d Gabor transforms by neural networks for image analysis and compression. IEEE Transactions on Acoustics, Speech and Signal Processing 36(7), 1169–1179 (1988)
8. Halfhill, T.R.: Parallel processing with cuda. IN-STAT Microprocessor Report, 1–8 (January 2008)
9. Hubel, D.H., Wiesel, T.N.: Receptive fields of single neurones in the cat's striate cortex. J. Physiol. 148, 574–591 (1959)
10. Hubel, D.H., Wiesel, T.N.: Ferrier Lecture functional architecture of macaque visual cortex. Proc. R. Soc. Lond. B. 198, 1–59 (1977)
11. Lourens, T.: A Biologically Plausible Model for Corner-based Object Recognition from Color Images. Shaker Publishing B.V., Maastricht (1998)
12. Lourens, T.: Tivipe –tino's visual programming environment. In: IEEE COMPSAC 2004, pp. 10–15 (2004)
13. Lourens, T., Barakova, E.I.: Tivipe simulation of a cortical crossing cell model. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 122–129. Springer, Heidelberg (2005)
14. Lourens, T., Barakova, E.I.: Orientation contrast sensitive cells in primate v1 –a computational model. Natural Computing 6(3), 241–252 (2007)
15. Lourens, T., Barakova, E.I., Okuno, H.G., Tsujino, H.: A computational model of monkey cortical grating cells. Biological Cybernetics 92(1), 61–70 (2005)

16. Lourens, T., Barakova, E.I., Tsujino, H.: Interacting modalities through functional brain modeling. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2686, pp. 102–109. Springer, Heidelberg (2003)
17. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of Imaging understanding workshop, pp. 121–130 (1981)
18. Sobel, I.E.: Camera Models and Machine Perception. PhD thesis, Electrical Engineering Department, Stanford University, Stanford, CA (1970)
19. Würtz, R.P., Lourens, T.: Corner detection in color images through a multiscale combination of end-stopped cortical cells. Image and Vision Computing 18(6-7), 531–541 (2000)
20. Zeki, S.: A Vision of the Brain. Blackwell science Ltd., London (1993)

# Using Biologically Inspired Visual Features and Mixture of Experts for Face/Nonface Recognition

Zeinab Farhoudi[1] and Reza Ebrahimpour[2,3]

[1] Department of Computer Engineering, Islamic Azad University Science
and Research Branch, Tehran, Iran
zeinabf22@gmail.com
[2] School of Cognitive Sciences, Institute for Studies on Theoretical Physics
and Mathematics, Niavaran, Tehran, P.O.Box 19395-5746, Iran
[3] Department of Electrical and Computer Engineering, Shahid Rajaee University,
Tehran, Iran
ebrahimpour@ipm.ir

**Abstract.** This paper introduces a novel, effective applicability of features inspired by visual ventral stream and biologically-motivated classification model, mixture of experts network for face/nonface recognition task. It describes a feature extracting system that derives from a feedforward model of visual cortex and builds a set of pose, facial expression, illumination and view invariant C1 features from all images in the dataset. Also, mixture of MLP experts network is a classifier which demonstrates high generalization capabilities in many different tasks. In accordance to these biological evidences, we propose face/nonface recognition model which combine these two techniques for the robust face/nonface problem. Experimental results using the combination C1 features and mixture of MLP experts network classifier, obtains higher recognition rate than related works in face/nonface identification. In addition, experimental results demonstrate this method is illumination and view-invariant.

**Keywords:** C1features; Eigenfaces; Face/Nonface recognition; Inferotemporal cortex; Mixture of Experts.

## 1 Introduction

Face/Nonface Recognition is one of the most attractive tasks since humans visual system is the only example of system which can perform identification task easily at high level of accuracy. The popular approaches have been focused on correlation or template matching, match filtering, subspace methods, deformable templates, eigenfaces and etc. However, the popular approaches emphasize on machine learning techniques and statistical modeling to discriminate between patterns found in natural images [1], it still remains unclear what method extract the best feature sets. Some methods to detect faces are the probabilistic visual learning method [2], the example–based learning method presented by Sung and Poggio [3] which cluster face and nonface samples into a few (i.e., 6) clustering using K-means algorithm, the neural network–based learning method which train multiple multilayer perceptrons with

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 439–448, 2009.

different receptive fields and rotation invariant face detection using neural network by Rowley [4,5], random graph matching, a feature based method which find the correct geometric arrangement of facial features [6], the machine learning approach using a boosted cascade of simple features [7] and the Bayesian discriminating feature (BDF) method [8].

In the late 1980s, successfully techniques experimented in human faces, was PCA (Principle Component Analysis) which efficiently represent human faces [9]. Given a set of different face images, PCA technique extracts eigenvectors from covariance matrix to find the principle component of the distribution of faces. Each individual face in the face set can then be approximated by a linear combination of the largest eigenvectors, more commonly referred to as eigenfaces, using appropriate weights. Turk and Pentland [10] later developed this technique for face recognition.

In recent research at ventral stream of primate visual cortex, different theories have been presented which interpret encoding information at different level of visual cortex and try to present a model. A core of fact about the ventral stream in the visual cortex is hierarchical processing. Along the hierarchy, the initial processing of information is feedforward. In a higher stage of processing, neurons in visual V4 area exhibit partially complex shapes with information about the structural description of the represented feature [11]. In the final stage of processing, neurons in inferotemporal cortex (IT) area, learning to recognize 3D object and they generate a corresponding activity according to face shape information. It is assume that V4 and IT areas play an important role for the detection and recognition of human faces [12]. Riesenhuber and Poggio propose a model that is motivated from a quantitative theory of ventral stream of visual cortex [13]. This system has high performance at complex scenes. In these scenes object recognition must be robust respect to pose, scale, position, rotation and Image condition (lighting, camera characteristics and resolution). The simplest version of the model consists of four layers that try to summarize a core of well-accepted facts about the ventral stream in the visual cortex [14].

Mixture of experts, MOE, first proposed in [15,16]. Their proposed model contains a population of simple linear classifier (the experts) whose outputs are mixed by gating network. The experts are technically performing supervised learning, in the other words they self-organize to find a good partition of the input space. Jordan and Jacobs [17] extended the model to so-called “hierarchical mixture of experts”, in which each component of MOE is replaced with a mixture of experts model.

In this paper to achieve a level of powerful classification, we combine face representations with mixture of multilayer perceptron experts network classifier. In this classifier, a modified structure for the expert networks is applied, which employs multilayer perceptrons, MLPs, to form the building blocks of the mixture architecture. By combining these results on face identification with the high performance previously shown for multilayer perceptron experts network, the current features representation model is starting to resemble a neurologically based computational system capable of approaching human performance on the ‘complete’ visual recognition problem.

The rest of this paper is organized as follows: Section 2 describes the proposed model of face/nonface recognition. Section 3 presents experimental results on face/nonface recognition and some discussions. Section 4 concludes the paper.

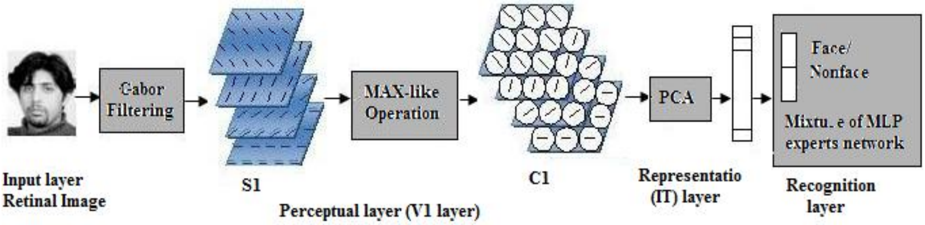


**Fig. 1.** System overview: The proposed system consists of three biological layers: perceptual analysis, object representation and recognition layer

## 2 The Proposed Model

We propose a new model based on biological visual feedforward pathways as shown in Figure 1. The first layer of the model quantitatively corresponds to the tuning properties of V1 cells along the ventral stream of visual cortex. This level is named the "Perceptual" layer. The first layer of the perceptual layer consists of two layers of computational units: simple S1 units and complex C1 units. A standard way to model S1 cells in other visual recognition networks is to use Gabor filters [18]. These units do nonlinear edge detection by convolving an array of Gabor filters at four orientations and ten scales, over the input image. Pairs of S1 units at adjacent scales are then grouped together to create 5 'bands' of units for each of the orientations. The C1 layer is then created by taking the maximum response within a local spatial neighborhood and across the scales within a band, to create a representation that contains 5 bands × 4 orientations. This means that the first layer of the model gives responses that are robust to small changes in the image.

In order to avoid a high dimensional and redundant input space and optimally design and train the expert networks, using a technique to reduce the input space dimensions is necessary. To solve the problem of high dimensional input space and transform features into eigenfaces, PCA is used which is a biologically plausible as is the population of so–called "face cells" in inferior temporal cortex [19]. This layer is termed the "Representation" layer.

On the other hand, there are biological evidences indicating the existence of IT neurons which completely separate the input classes by extracting orthogonal dimensions from the data [20]. The implementation of PCA method is as the following steps: first normalizing the data, second calculating the covariance matrix; third calculating the eigenvectors and eigenvalues of the covariance matrix, and then choosing components and forming a feature vector. The representation layer's outputs are then identified as two "faces and nonfaces" classes by a mixture of multilayer perceptron experts network that is a supervised modular neural network. This layer is termed the "Recognition" layer.

Inside the recognition layer, an efficient face/nonface recognition method is build which uses a type of mixture of multilayer perceptron experts architecture shown in Figure 2. Each classifier gives a decision on which subspace each input space belongs to, and their outputs are combined through the gating network to produce the final decision. In addition to its robustness and the performance of the model to handle a larger face/nonface space, evidence for this kind of structures of neurobiology and neuroimaging plausible can be found in [21]: "…Indeed, it is hard to imagine how this could occur without a special gating mechanism that discriminates between faces and nonfaces and allows only the face information into the FFA…".

## 2.1 Perceptual Layer

This layer corresponds to first two layers of primate ventral stream of visual cortex. C1 SMF system consists of two hierarchical layers of computational units composed of simple S1 units and complex C1 units [12] [13]. This layer of our framework is mostly based on that of Serre and Poggio et al. [14], with small changes according to the base model which consist of four hierarchical layers of computational units composed of alternating simple S units and complex C units. The S units combine their inputs with a bell-shaped tuning function to increase selectivity and the C units pool their inputs through a maximum (MAX) operation, thereby increasing invariance. This layer is summarized as follow.

**S1 Units:** A gray-value input image is first analyzed by a multidimensional array of simple S1 units which correspond to the classical simple cells in the primate visual cortex (V1) [14]. At first stage, S1 responses are obtained by applying a battery of Gabor filters to the input images, which have been shown to provide a good model of cortical simple cell receptive fields. Indeed, the S1 units extract features with a localized, oriented-edge detection on the image, with help of 2D Gabor filter, that react to line of a particular orientation, scale, and position. Gabor function can be described by the following equation:

$$F\,(x,y) = \exp\left(-\frac{\left(x_0^2+\gamma^2 y_0^2\right)}{2\sigma^2}\right)\times\cos\left(\frac{2\pi}{\lambda}x_0\right). \quad (1)$$

$$x_0 = xcos\theta + ysin\theta. \quad y_0 = -xsin\theta + ycos\theta. \quad (2)$$

where the aspect ratio $\gamma$=0.3, the orientation $\theta$ and the effective width $\sigma$ , the wavelength $\lambda$ were adjusted so that the tuning calls of the corresponding S1 units match those of V1 parafoveal simple calls as closely as possible. In our approaches, we arranged the S1 filters to form a pyramid of scales, spanning a range of sizes from 7×7 to 25×25 pixels in steps of two pixels and considered four orientations (0°,45°,90°,135°) , thus leading to 40 different S1 receptive field types (10 scales×4 orientations), and the C1 layer and S1 layer parameters have been arranged as [14].

**C1 Units:** The next C1 level corresponds to cortical complex cells which illustrate some tolerance to shift and size. Complex cells tend to have larger receptive field and to be more tuned than simple cells. C1 units pool over afferent S1 units from the previous layer by performing nonlinear MAX-like operation over the same orientation within the neighborhood of S1 units in both space and scale. In other words, the response r of a complex unit is the strongest response of its m afferents (x1,…,xm) from the previous S1 layer such that:

$$R = \max_{i=1\ldots m} x_i. \quad (3)$$

Consider, each scale band contains two adjacent filter sizes (there are five scale bands for a total of 10 S1 filter sizes). For instance, the first scale band for each orientation contains two S1 filters with sizes 7×7 and 9×9. Note that both S1 filters have the same dimensions. The response of the C1 unit is obtained by sub-sampling these S1 maps using a cell grid of size $N_s \times N_s = 8\times8$. From each grid cell, the maximum of all 40 elements gives one single measurement. As a last stage, by taking a maximum over the two scales, for each call consider the maximum value from the two maps. Again, this process is independently repeated for each of the four orientations and each scale band.

### 2.2 Representation Layer

A standard technique which used to approximate the original data with lower dimensional feature vectors is Principal Component Analysis (PCA). The main idea of PCA is to find the appropriate vectors that can describe the distribution of face images in image space and form low dimensional subspace. Let the training set of face images be $\Gamma_1, \Gamma_2, \Gamma_3, ..., \Gamma_M$. The average face of the set is defined by $\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_{\mathbf{n}}$. Each face differs from the average by the vectors $\Phi = \Gamma_i - \Psi$. The eigenvalues and eigenvectors are obtained from covariance matrix (see Eq. (4)) and a new face image is transformed into its eigenface components by this operation [10].

$$C = \frac{1}{M}\left[\sum_{n=1}^{M}\Phi_{\mathbf{n}}\Phi_{n}^{T}\right] \tag{4}$$

The weights form a vector $\Omega = [w_1, w_2, \ldots w_M]$ that describes the contribution of each eigenvector in representing the input face image, treating the eigenvector as a basis set for face images.

$$w_k = u_k^T(\Gamma - \Psi), k = 1, 2, \ldots, M \tag{5}$$

where $u_k$ is eigenvector. These weights may be used in a face classification algorithm to find which of a number of predefined face classes that best describes the face.

### 2.3 Recognition Layer

*Mixture of multilayer perceptron experts architecture,* is the most famous method in the category of dynamic structure of classifier combining [22]. In our modified version of mixture of experts model we use MLPs, instead of linear networks as shown in Figure 2. In order to select the expert network with the best performance at solving the problem, a gating network simultaneously trained is used. Then the learning algorithm is corrected by using an estimation of the posterior probability of the generation of the desired output by each expert. Using this new learning method, the MLP expert networks' weights are updated on the basis of those estimations and the procedure is repeated for the training data set. Each expert network is an MLP network with one hidden layer that computes an output vector $O_i$ as a function of the input stimuli vector, x, and a set of parameters such as weights of hidden layer and output layer and a sigmoid activation function. We suppose that each expert specializes in a different area of the input space. The gating network assigns a weight $g_i$ to each of the experts' outputs, $O_i$. The gating network determines the $g_i$ as a function of the input vector x and a set of parameters such as weights of the hidden layer, a sigmoid activation function and the output layer. The $g_i$ can be interpreted as estimates of the prior probability that expert *i* can generate the desired output *y*. The gating network is composed of two layers: an MLP network and a Softmax nonlinear operator as the gating network's output. Therefore the output of the MLP layer of the gating network, $O_g$, is computed by the gating network, then the Softmax function is applies to get $g_i$:

$$g_i = \frac{\exp(O_{g_i})}{\sum_{j=1}^{N} \exp(O_{g_j})}, \qquad i = 1, \dots , N \tag{6}$$

where N is the number of expert networks. So the $g_i$ are nonnegative and sum to 1. The final mixed output of the complete network is

$$O_T = \sum_i O_i g_i, \qquad i = 1, \dots , N \tag{7}$$

In order to maximize the log likelihood of the training data given the parameters, the weights of MLPs are updated using the back–propagation (BP) algorithm, by the following equation:

$$\begin{aligned} \Delta w_y &= \eta_e h_i (y - O_i)(O_i(1 - O_i)) O h_i^2 \\ \Delta w_h &= \eta_e h_i w_y^T (y - O_i)(O_i(1 - O_i)) O h_i (1 - O h_i) x_i \end{aligned} \tag{8}$$

$$\begin{aligned} \Delta w_{yg} &= \eta_e (h - g)(O_g(1 - O_g)) O h_g^T \\ \Delta w_{hg} &= \eta_e w_{yg}^T (h - g)(O_g(1 - O_g)) O h_g (1 - O h_g) x_i \end{aligned} \tag{9}$$

where $\eta_e$ and $\eta_g$ are the learning rate for the expert networks and gating network, respectively. $Oh_i$ and $Oh_g$ are the output of the hidden layer of experts network and the gating network, respectively and $h_i$ is an estimate of the posterior probability that expert *i* can generate the desired output *y* (Eq. 10):

$$h_i = \frac{g_i \exp(-\frac{1}{2}(y - O_i)^T (y - O_i))}{\sum_j g_j \exp(-\frac{1}{2}(y - O_j)^T (y - O_i))} \tag{10}$$

the posterior probability, $h_i$, is considered as a Softmax function computed on the inverse of the sum squared error of each expert's output, smoothed by the gating network's current estimate of the prior probability that the input pattern was drawn from expert *i*'s area of specialization. As the network's learning process progresses, the expert networks "compete" for each input pattern, while the gating network gives a reward to the winner of each competition with stronger error feedback signals. Thus, over time, the gating network partitions the input space in response to the expert's performance.

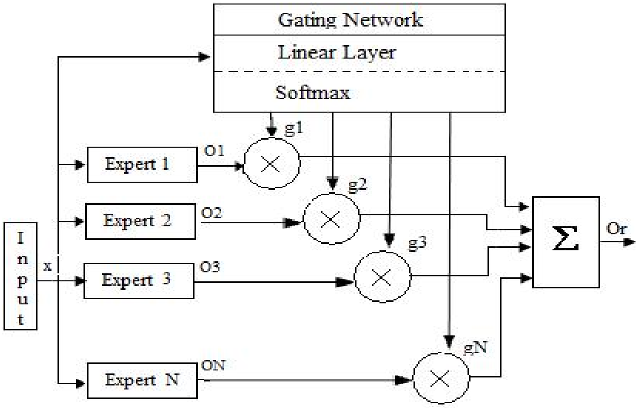


**Fig. 2.** Mixture of experts network [23]

## 3 Experimental Results

For the training phase, we used a set of 2700 face images collected from ORL, UMIST, Harvard, Yale and MIT CBCL databases and from the Internet, cropped and resized to 40×40 pixels which have wide variations in pose, facial expression and lighting conditions (Figure 3). We used a set of 2700 nonface images which are obtained by sampling from images containing landscapes, trees, buildings, cars, rocks, flowers etc. In the preprocessing step all images are scaled to a 40×40 size and converted to gray value. Then these images are first analyzed by an array of S1 units at four different orientations and 10 scales. At the next C1 layer, maximum operation is applied over a neighborhood of S1 units in both space and scale, but with the same preferred orientation. After the C1 features have been extracted, we have 20 (5 bands×4 orientations) face patterns from a 1600 dimensional image space. At the representation stage, the M_PCA matrix projects face patterns to a 50-dimensional subspace.

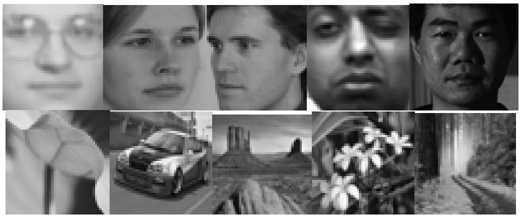


**Fig. 3.** Sample Images from both face and nonface categories

After extracting features from the face images, using a powerful classification structure is an important problem. At the recognition layer two pattern classifications are applied to recognize input images as face or nonface. The first method is mixture of multilayer experts network and the other is a complex MLP network. To evaluate

**Table 1.** Face/nonface recognition of testing the MLP and mixture of experts networks on the test set

| Network | Topology | Output Node Threshold Value | Learning rate | Training Epoch | Recognition Rate (%) |
|---|---|---|---|---|---|
| MLP [30] | 50:85:1 | 0.49 | 0.05 | 950 | 79.8 |
| Mixture of Experts [30] | Gating<br>50:15:2 | 0.476 | $\eta_e$ =0.054 | 450 | 97.6 |
| | Experts<br>50:20:1 | | $\eta_g$ =0.014 | | |
| C1 + MLP | 50:85:1 | 0.4 | 0.02 | 600 | 94.5 |
| C1+ Mixture of MLP | Gating<br>50:15:2 | 0.4 | $\eta_e$ =0.054 | 450 | 98 |
| | Experts<br>50:20:1 | | $\eta_g$ =0.014 | | |

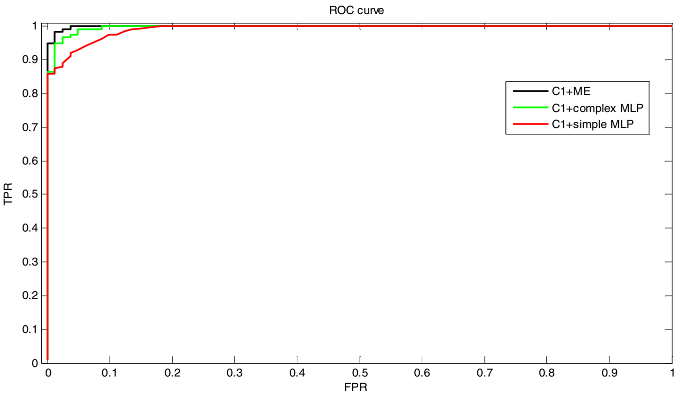


**Fig. 4.** ROC curves illustrating the performance of Mixture of MLP with C1 features is much better in comparison with other networks [24]

the performance of mixture of multilayer experts network, we compare these two methods. The experimental results illustrate the capability of mixture of experts network at discriminating patterns is much better than complex MLP in many ways (see Table 1). The mixture network model consists of two simple MLPs for experts and one simple MLP for gating network and the complex network consist of a three-layer MLP with one hidden layer.

We tried training and testing of different MLP topologies and we find the best structure with 50:85:1 for complex MLP. The detailed of our experiment is reported in Table 1. The ROC curves of MLP and the mixture of multilayer perceptron experts network of our model compared to [24] are plotted in Figure 4. It is obvious that a mixture of MLP experts with C1 features is much better than other network topologies.

## 4 Conclusion

Understanding a neuroscience model of visual cortex performs object recognition is one of the ultimate goals in computational biology. Recognition of face/nonface is a difficult problem, which confronts all the major challenges in computer vision and pattern recognition. This paper presented a biologically-motivated framework for face/nonface recognition. First C1 maps were computed for each image and then the discriminative structure of mixture of experts network run. Since the training images have only changes in pose, illumination and view, using C1 SMF is suitable for the face represent. The experiments results demonstrate the proposed approach is powerful in extracting the relevant discriminatory information from the training face data and the discriminate structure of mixture of MLP experts is very high efficiency for the test C1 features. In the face/nonface recognition task the performance of the mixture of two simple MLP comparisons with a complex MLP reveals that a mixture structure is much more reliable at solving complex pattern recognition problem. Also, it has the advantage of easier training, because of simpler expert networks.

## References

1. Yang, M., Kriegman, D.J., Ahuja, N.: Detecting Faces in Images: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1), 34–58 (2002)
2. Moghaddam, B., Pentland, A.: Probabilistic visual learning for object representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 696–710 (1997)
3. Sung, K.K., Poggio, T.: Example–based learning for view–based human face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(1), 39–51 (1998)
4. Rowley, H.A., Baluja, S., Kanade, T.: Neural network–based face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(1), 23–38 (1998)
5. Rowley, H.A., Baluja, S., Kanade, T.: Rotation invariant neural network–based face detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, USA, pp. 38–44 (1998)
6. Leung, T.K., Burl, M.C., Perona, P.: Finding Faces in Cluttered Scenes Using Labeled Random Graph Matching. In: ICCV, pp. 637–644 (1995)

7. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Kauai, Hawaii, pp. 511–518 (2001)
8. Liu, C.: A Bayesian discriminating features method for face detection. Transaction on Pattern Analysis and Machine Intelligence 25(6), 725–740 (2003)
9. Sirovich, L., Kirby, M.: Low–dimensional procedure for the characterization of human faces. Journal of Optical Society of America 4, 519–524 (1987)
10. Turk, M., Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neuroscience 3(1), 71–86 (1991)
11. Pasupathy, A., Connor, C.E.: Population coding of shape in area V4 Nature. Neuroscience 5(12), 1252–1254 (2002)
12. Riesenhuber, M., Poggio, T.: Hierarchical Models of Object Recognition in Cortex. Nature Neuroscience 2(11), 1019–1025 (1999)
13. Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., Poggio, T.: A Theory of Object Recognition: Computations and Circuits in the Feedforward Path of the Ventral Stream in Primate Visual Cortex, AI Memo 2005-036/CBCL Memo 259. Massachusetts Inst. of Technology, Cambridge (2005)
14. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust Object Recognition with Cortex-like Mechanisms. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(3), 411–426 (2007)
15. Jacobs, R., Jordan, M., Barto, A.: Task decomposition through competition in a modular connectionist architecture: what and where vision tasks. Tech. rep. University of Massachusetts, Amherst, MA (1991)
16. Jacobs, R., Jordan, M., Nowlan, S., Hinton, G.: Adaptive mixtures of local experts. Neural Computer 3, 79–87 (1991)
17. Jordan, M., Jacobs, R.: Hierarchical mixtures of experts and the EM algorithm. Neural Computer 6(2), 181–214 (1994)
18. Lades, M., Vorruggen, J.C., Buhmann, J., von der Lange, J., Malsurg, C., Wurtz, R.P., Konen, W.: Distortion invariant object recognition in the dynamic link architecture. IEEE Transactions on Computers 42(3), 300–311 (1993)
19. Perrett, D., Hietanen, J., Oram, M., Benson, P.: Organization and functions of cells responsive to faces in the temporal cortex. Philosophical Transactions of the Royal Society of London Series BBiological Sciences 335, 23–30 (1992)
20. Kayaert, G., Biederman, I., Vogels, R.: Representation of Regular and Irregular Shapes in Macaque Inferotemporal Cortex. Cerebral Cortex 15, 1308–1321 (2005)
21. Kanwisher, N., Downing, P., Epstein, R., Kourtzi, Z.: Introduction: the modularity of visual recognition. In: Handbook of Functional Neuroimaging of Cognition, pp. 109–151. MIT Press, Cambridge (2001)
22. Haykin, S.: Neural networks: a comprehensive foundation. Prentice Hall, USA (1999)
23. Ebrahimpour, R., Kabir, E., Yosefi, M.R.: Face detection using mixture of MLP experts. Neural Procassing Letters 26, 69–82 (2007)
24. Ebrahimpour, R., Kabir, E., Estekey, H., Yosefi, M.R.: A mixture of multilayer perceptron experts network for modeling face/nonface recognition in cortical face processing regions. Intelligent Automatic and Soft Computing 14(2), 145–156 (2008)

# Diagnosis Support System for Mucous Membrane Diseases in Oral Cavity

Keiichi Horio[1], Shuhei Matsumoto[1], Taishi Ohtani[2],
Manabu Habu[2], Kazuhiro Tominaga[2], and Takeshi Yamakawa[1]

[1] Grad. Sch. of Life Science and Systems Engineering, Kyushu Inst. of Tech.,
2-4 Hibikino, Wakamatsu, Kitakyushu, 808-0196, Japan
[2] Dept. of Oral and Maxillofacial Surgery, Kyushu Dental College,
2-6-1 Manazuru, Kokura-kita, 803-8580, Japan
horio@brain.kyutech.ac.jp
http://www.brain.kyutech.ac.jp/~horio/

**Abstract.** In this paper, a diagnosis support system for mucous membrane diseases in oral cavity, in which a shape of vitiligo is judged as lacy or not-lacy from an intraoral image, is proposed. Intraoral images are not unstandardized, and lighting conditions or angles are variant. To extract vitiligo area from such images, we employ dynamic threshold method. SIFT and bag of keypoints are used in feature extraction, and classification is achieved by SVM. From the simulation, in which the shape of vitiligo is classified to lacy or not-lacy classes, the effectiveness of the proposed system was verified.

**Keywords:** diagnosis support system, mucous membrane disease in oral cavity, dynamic threshold.

## 1 Introduction

There are many patients with oral mucosal diseases, however, diseases which are considered as serious ones are limited such as squamous cancer, leukoplakia, lichen planus in clinical practice. It is not difficult for surgeon dentist to discriminate these diseases, but it sometimes becomes difficult for general dentist to do that. As the digital camera spreads in recent years, number of general dental office that intraoral images used for daily clinical examination increases. Diagnosis of oral mucosal diseases is achieved mainly based on visual inspection, and the specialists can discriminate them with high accuracy even from intraoral images. This fact is our motivation for constructing a diagnosis support system for oral mucosal diseases based on intraoral images.

Fig. 1 shows a classification tree constructed based on interviews of some specialists. Our goal is to realize a discrimination system which performs like the decision tree shown in Fig. 1. As a first step, we discuss "Is vitiligo lacy?" in this paper. In this discrimination, the important information is shape of vitiligo. In case of leukoplakia, the vitiligo spreads evenly. On the other hand, the vitiligo spreads lacily in case of lichen planus. Leukoplakia is an oral precancerous lesion

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 449–456, 2009.

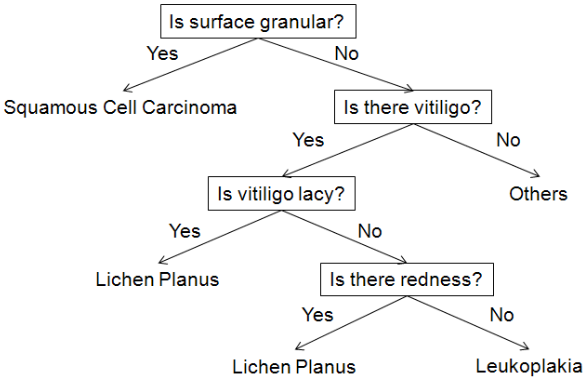


**Fig. 1.** Classification tree

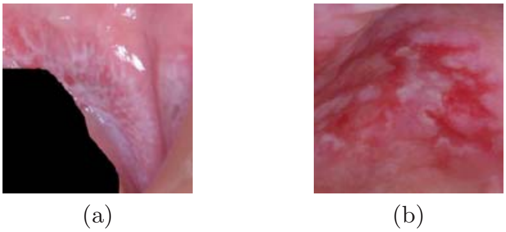


**Fig. 2.** Examples of intraoral images with vitiligo. (a) Lacy vitiligo and (b) not-lacy vitiligo.

(possibility to be cancerous is 3-10%), and it is strongly required to find out it in early stage. Fig. 2 (a) and (b) show examples of lacy vitiligo and not-lacy vitiligo judged by the specialists, respectively. The objective of this paper is to classify the intraoral images to two classes based on the shape of vitiligo, *i.e.* lacy or not.

## 2 Overview of the Proposed Method

When intraoral images are taken in general clinical practice of dentistry using commercially digital cameras, lighting condition and angles are unstandardized. Color information on diseased part varies widely, and halation is sometimes observed. These cause classification to be difficult.

To cope with this problem, extraction of vitiligo area is employed as a preprocessing. Color of vitiligo area is not fixed, it changes according to conditions, inidividuals, and so on. In other words, Color difference between vitiligo and other areas is relative. Furthermore, the color is different from one locality to another even in same image. It is needed to ectract vitiligo using only local information. In lacy vitiligo image, there are many local features which include white curvatures. To represent these type of local features, scale invariant feature transformation (SIFT),

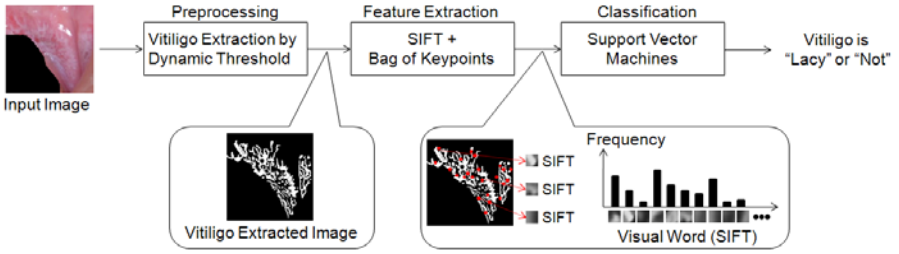


**Fig. 3.** Block diagram of the proposed classification method

wchich is robust for scaling, shifting, rotating, is usefull. As a clasifier, support vector machine (SVM), which is one of the most powerful classifier, is adopted. The block diagram of the proposed classification method is shown in Fig. 3.

## 3 Extraction of Vitiligo

To extract vitiligo area from a intraoral image is considered as to binarize the image. In the binarized images, white and black pixels mean vitiligo and not-vitiligo, respectively. The simplest method to binarize images is to use threshold. As mentioned above, however, color on vitiligo in intraoral image includes wide variations caused by difference of condition of lighting and angle. Some pixels of vitiligo area are whitish, and some pixels are red-tinged even in one image. In such case, binarization with fixed threshold is not useful, because the pixels of vitiligo and not-vitiligo are overlapped in a color space. As binarization methods, in which the threshold is determined by considering color distribution, percentile method, mode method, discriminant analysis are well known[1][2][3]. In these methods, however, thresholds are determined in consideration of color distribution of whole image, and it is not effective when one image includes light and dark areas like this problem.

On the other hand, surgeon dentists judge the pixel as vitiligo or not using only local image around focused pixel. One of the simplest methods is to use a threshold decided based on pixel distribution of local image. In such case, however, it is difficult to decide the size of local image, and it requires much computational cost. Thus we employ a dynamic threshold method[4][5].The algorithm of the dynamic threshold is as follow.

1. Original color image is transformed to gray scale image.
2. Gray scale images are smoothed by using smoothing filter (Gaussian filter).
3. Difference image between gray scale image and smoothed image is obtained. Each pixel is decided to white or black if the pixel value is larger or smaller than threshold.
4. Obtained binary image is smoothed to reject noise.

The size of Gaussian filter and threshold are parameters to be decided.

## 4 Feature Extraction

In this study, an objective is that shape of vitiligo is judged as lacy or not. It is reasonable that we assume lacy vitiligo includes many edges in the image. Thus the feature histogram which represents a proportion of various features included in the image is employed as feature of the image.

### 4.1 SIFT

SIFT is a feature extraction algorithm proposed by Lowe[6][7]. SIFT is robust for scaling, rotating and change of lighting condition, and its algorithm is divided into two parts, keypoint extraction and feature representation.

In the keypoint extraction, the candidates of keypoints are firstly extracted by processing called Difference of Gaussian (DoG). DoG image is defined by

$$DoG(u, v, \sigma_i) = L(u, v, \sigma_{i+1}) - L(u, v, \sigma_i), \tag{1}$$

$$L(u, v, \sigma_i) = G(x, y, \sigma_i) * I(u, v), \tag{2}$$

$$G(x, y, \sigma_i) = \frac{1}{2\pi\sigma_i^2} \exp(-\frac{(x^2 + y^2)}{2\sigma_i^2}), \tag{3}$$

where, $I(u, v)$ and $G(x, y, \sigma_i)$ are intensities of coordinate $(u, v)$ of original image and cooefficient of coordinate $(x, y)$ of Gaussian filter, and $\sigma_i$ is width of the filter. As shown in above, in DoG, color substraction between images smoothed by Gaussian function with different variances is calculated, and candidates can be determined based on this color substraction. Then the candidates which are susceptible to the effect of noise or aperture problem are removed.

In feature representation, 128 dimensional feature vector for each keypoint is detected. At first orientation which is main direction of color gradient around the keypoint is determined. And local area are divided into 16 (4×4) subareas in consideration of the orientation. For each subarea, gradient histogram for eight directions is obtained. As a result, 128 dimensional feature vector is calculated for each keypoint. Please refer [6][7] for detail.

### 4.2 Bag-of-Keypoints

Bag-of-keypoints (BoK)[8] is a model which represents image as a set of local features, here, positions of features are not considered. In fact, histogram of local features is used as a feature of the image. BoK can be considered as one of the codebook, and codes in codebook are called as visual words in BoK. Ability of classification of the images with BoK is based on number and selection of visual words. In this study visual words correspond to SIFT. Generally, appropriate number of visual words depend on type of objects to be classified. There is however, no knowledge for this application, and the number of visual word $k$ is decided experimentally. A $k$-means algorithm for SIFT extracted from images to determin the visual words.

# 5 Classification by SVM

As mentioned above, each image is represented by histogram of local features. Classifying the images to lacy-vitiligo or not-lacy-vitiligo is achieved by Support Vector Machine (SVM)[9]. SVM is one of the most powerful classifiers, and is widely applied to many applications. In SVM, kernel is used to transform the patterns to high dimensional feature space, and nonlinear classification in original input space can be considered as linear classification in high dimensional feature space by presenting the kernel method. In this study, polynomial kernel

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (1 + \boldsymbol{x}_1^T \boldsymbol{x}_2)^p \tag{4}$$

is used. Many types of kernel are well known, polynomial kernel is experimentally chosen. In the simulation presented in the next section, $p$ is 10.

# 6 Simulation Results

## 6.1 Data Used in Simulation

Images used in this study are taken by dentists with commercial digital cameras, and 19 images (14 for lacy and 5 for not-lacy) are prepared. Some images include teeth, and teeth are preliminarily removed by using photo-retouching software. To evaluate the simulation results, binary images in which vitiligo area is manually extracted by the surgeon dentist is also prepared, and these images are called teacher images.

## 6.2 Vitiligo Extraction Simulation

**Binarization by Fixed Threshold.** Threshold is decided from the distribution of all pixels of all images in color space (YCrCb). All pixels can be labeled as vitiligo or not based on teacher images. Of course, distributions of two classes are overlapped in color space. Probabilities that a new pixel $\boldsymbol{x}$ belongs to vitiligo or not are defined by Eq.(5) and Eq.(6), respectively.

$$p_v(\boldsymbol{x}) = \frac{q_v(\boldsymbol{x})}{q_v(\boldsymbol{x}) + q_n(\boldsymbol{x})}, \tag{5}$$

$$p_n(\boldsymbol{x}) = \frac{q_n(\boldsymbol{x})}{q_v(\boldsymbol{x}) + q_n(\boldsymbol{x})} \tag{6}$$

where,

$$q_v(\boldsymbol{x}) = \exp(-\frac{\parallel \boldsymbol{x} - \boldsymbol{\mu}_v \parallel}{2\sigma_v^2}), \tag{7}$$

$$q_n(\boldsymbol{x}) = \exp(-\frac{\parallel \boldsymbol{x} - \boldsymbol{\mu}_n \parallel}{2\sigma_n^2}), \tag{8}$$

where, $\boldsymbol{\mu}_v$ and $\boldsymbol{\mu}_n$ are means vector of pixels of vitiligo and non-votiligo, respectively, and $\sigma_v$ and $\sigma_n$ are standard deviations of pixels of vitiligo and non-votiligo, respectively. Finally the new pixel is judged as vitiligo or not by maximum probabilities.

**Binarization by Dynamic Threshold.** Ability of binarization by dynamic threshold is based on parameters, size of mask and threshold. These parameters are experimentally decided to maximize an evaluation function[10] represented by:

$$E = \frac{1}{X \times Y}(1 - \frac{\sum_{y=1}^{Y}\sum_{x=1}^{X} w(x,y)|t(x,y) - p(x,y)|}{\sum_{y=1}^{Y}\sum_{x=1}^{X} w(x,y)r}), \tag{9}$$

where, $X$ and $Y$ are horizontal and vertical size of a image, respectively, and $t(x,y)$ and $p(x,y)$ are intensities of pixel $(x,y)$ of teacher and output images, respectively. The evaluation function means the error between teacher and output images weighted by $w(x,y)$, and $r$ is resolution for intensity (in this case $r = 256$). The weight $w(x,y)$ is 255 and 200, when the pixel of the teacher image is white and black, respectively. As a result, size of mask is 85, and threshold is 7.

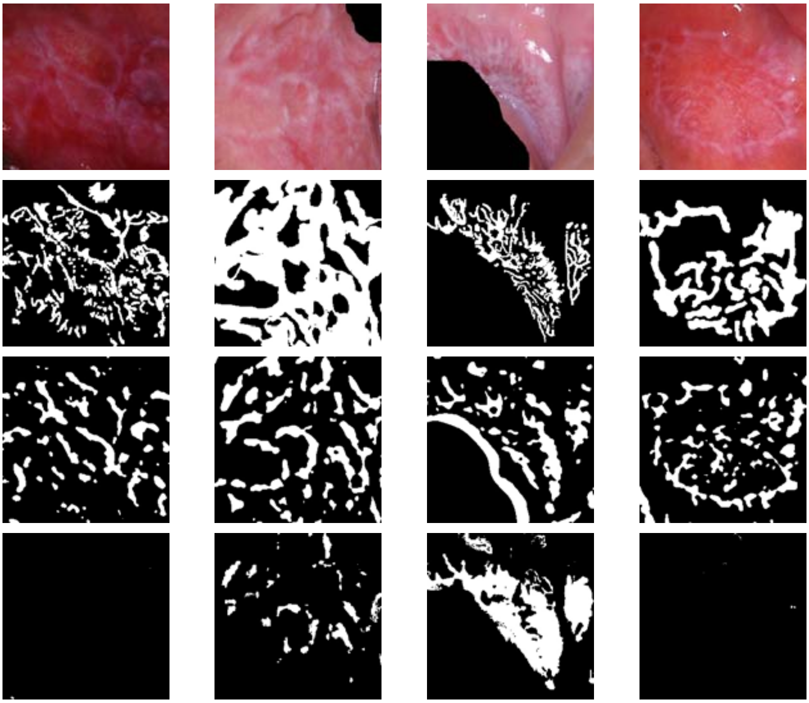


**Fig. 4.** Examples of binarization of four images. From top to bottom, original color image, teacher image, binarized by dynamic threshold, and binarized by fixed threshold.

Fig.4 shows examples of binarization of four images. Original color images, teacher images, images binarized by dynamic threshold and images binarized by fixed threshold are indicated from top to bottom of Fig.4.

### 6.3 Classification Simulation

In classification simulation, we employ cross-validation for verifying an effectiveness of the proposed method. The visual words of the BoK and the parameters of SVM are tuned using 18 images, and remaining one image is used for testing, that is the remaining image is classified by SVM trained using 18 images. The processes above are achieved for all the images, and a classification ratio is obtained. Fig. 5 shows the classification ratios corresponding to number of visual words $k$. For comparison, four methods are employed for preprocessing, dynamic threshold (green), fixed threshold (red), original image, *i.e.* no preprocessing (pink) and teacher image (blue). It is shown that the teacher image gives the best result, and it means that the surgeon dentist adequately extract the vitiligo area and judge that vitiligo is lacy or not. The result of fixed threshold is worse than that of original image. It is considered that information amount which represents features in the image is reduced by inadequate binarization. On the other hand, dynamic threshold presents good result, and the classification ratio is almost same to that of teacher image, when number of visual words is appropriately decided. It means that vitiligo is extracted with high accuracy by employing the dynamic threshold algorithm.

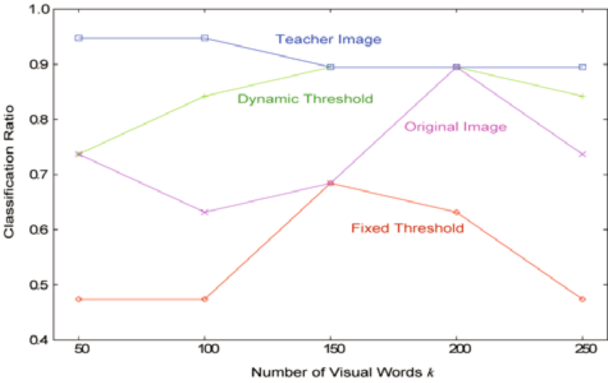


**Fig. 5.** Classification ratio

## 7 Conclusions

In this paper, the diagnosis support system for mucous membrane diseases in oral cavity was proposed. In the proposed system, vitiligo area was adequately extracted from intraoral image even when the lighting condition and angle changed.

And it was confirmed that the shapes of vitiligo were successfully classified to two classes, lacy or not. From the results, vitiligo extraction by the dynamic threshold, in which the threshold is determined in consideration of only local area, mimic extraction process of the surgeon dentist.

## References

1. Sahoo, P.K., Soltani, S., Wong, A.K.C., Chen, Y.C.: A Survey of Thresholding Techniques. Computer Vision, Graphics, and Image Processing 41, 233–260 (1988)
2. Weszka, J.S.: A Survey of Threshold Selection Techniques. Computer Graphics and Image Processing 7, 259–265 (1978)
3. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. IEEE Trans. on Sys., Man, and Cybern. 9, 62–66 (1979)
4. Chow, C.K., Kaneko, T.: Automatic Boundary Detection of the Left Benticle from Cineagrams. Computer and Biomedical Res. 5, 863–873 (1988)
5. Matsushima, H., Yamachiro, T., Toguchi, M.: Simple and Fast Binarization Technique which is used to background density distribution. In: Report of the 200th Technical Conf. on the Inst. of Image Elect. Eng. of Japan, pp. 169–175 (2003) (in Japanese)
6. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. of Computer Vision. 60, 91–110 (2004)
7. Lowe, D.G.: Object recognition from local scale-invariant features. In: Int. Conf. Comp. Vis., pp. 1150–1157 (1999)
8. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: European Conference on Computer Vision Workshop on Statistical Learning in Computer Vision, pp. 1–22 (2004)
9. Cristianini, N., John, T.: An Introduction to Support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
10. Aoki, S., Nagao, T.: Automatic Construction of Tree-structural Image Transformation. J. of the Institute of Image Information and Television Engineering 53, 888–894 (1999)

# Using Long and Short Term Memories in Solving Optimization Problems

Masahiro Nagamatu and Jagath Weerasinghe

Kyushu Institute of Technology, 2-4 Hibikino,
Wakamatsu-ku, Kitakyushu-shi, 808-0196, Japan
nagamatu@brain.kyutech.ac.jp,
jagath-w@edu.brain.kyutech.ac.jp

**Abstract.** In this paper, we propose an optimization method which is based on the Lagrangian method. Experimental results show that the search can effectively find the feasible solutions. We also introduce long and short term memories to the Lagrangian method. There are several possible ways of integrating the long and short term memories. We examine some of basic ways of integration in the experiments, and decide the effective one. When we use this effective way of integration, we can improve the efficiency of the Lagrangian method furthermore.

**Keywords:** Optimization, constraint satisfaction, local optima, long and short term memories.

## 1 Introduction

Solving optimization problems are important not only in the industrial and business fields but also in the field of artificial intelligence where agents are required to act intellectually and effectively. The former includes designing, manufacturing, controlling, and decision-making, and the latter includes robot navigation and planning. However, as denoted by the fact that almost all optimization problems are NP-hard, it is generally difficult to find an optimum solution in a huge solution space. The existence of the local optimal solutions makes the problem further difficult.

To overcome this difficulty, several optimization techniques use the strategies inspired by human kinds or animals. Using curiosity, long term memory and short term memory are examples of them [1][2][3][4][5]. In the field of psychology, human kinds have two types of curiosities, i.e., specific and diversive curiosity. In the optimization, the specific curiosity is used to examine promising parts of the solution space carefully, and the diversive curiosity is used to visit unexplored parts of the solution space looking for a new solution. The long and short term memories are used in a similar way. The long term memory is used to store the information acquired using long time intervals, which is used to find new and promising parts of the solution space, while the short term memory is used to store the information which is used to decide actions for the immediate future by reacting to the current situation. In the optimization, it is general that long term memory is used to store the information of already visited

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 457–464, 2009.

solutions, and is used to look for new solutions. The short term memory is used to escape from the current local optimal solution.

In this paper, we study about using long and short term memories in the Lagrangian method for solving an optimization problem in electronics system design, and propose a good way of integrating the long and short term memories.

## 2 Lagrangian Method for Optimization

We have been solving the satisfiability problem (SAT) of the propositional calculus by the neural network called LPPH (Lagrange programming neural network with polarized higher order connections) [4][5][6]. The network is not trapped at any points which are not the solutions of the SAT, and if the trajectory comes near a solution, it converges to that solution. The same techniques can be used to solve an optimization problem.

(Optimization problem)

$$\left|\begin{array}{l} \text{minimize} f(\boldsymbol{x}), \\ \text{subject to} \quad C_r, \quad r = 1,2,\ldots,m, \\ \qquad\qquad\quad x_i = 0,1, \quad i = 1,2,\ldots,n. \end{array}\right.$$

(Lagrangian method (LAG))

1) Make an initial solution $\boldsymbol{x}_0$. $\boldsymbol{w}_0=(w_{1,0},w_{2,0},..,w_{m,0})$. $t$=0.

2) Find a solution $\boldsymbol{x}_{t+1} = \underset{\boldsymbol{x}}{\text{argmin}} \left\{ F(\boldsymbol{x},\boldsymbol{w}_t) = f(\boldsymbol{x}) + \sum_r w_{r,t} h_r(\boldsymbol{x}) \mid \boldsymbol{x} \in N(\boldsymbol{x}_t) \right\}$.

3) $\boldsymbol{w}_{t+1}= \boldsymbol{w}_t - \alpha \boldsymbol{w}_t + \lambda h_r(\boldsymbol{x}_{t+1})$.

4) If $\boldsymbol{x}_{t+1}$ satisfies the terminating condition then halt, else $t$=$t$+1 and goto 2).

In the above definition, $f(\boldsymbol{x})$ is an objective function, $C_r$ is a constraint, $w_{r,t}$ is the weight (Lagrange multiplier) of constraint $C_r$, $F(\boldsymbol{x},\boldsymbol{w})$ is the Lagrangian function, $h_r(\boldsymbol{x})$ is a function which indicates the degree of unsatisfaction of constraint $C_r$ ( $h_r(\boldsymbol{x})\geq 0$, and $h_r(\boldsymbol{x})=0$ *iff* $C_r$ is satisfied ), $N(\boldsymbol{x})$ is a neighborhood of $\boldsymbol{x}$, $\alpha\geq 0$ is a constant called "decay factor", and $\lambda\geq 0$ is a constant called "cost coefficient".

We also introduced long and short term memories to the LPPH. In the same manner, we can introduce long and short term memories to the above Lagrangian method as follows.

(Lagrangian method with long and short memories (LAGLS))

1) Make an initial solution $\boldsymbol{x}_0$.
$\boldsymbol{w}_0=(w_{1,0},w_{2,0},..,w_{m,0})$. $\boldsymbol{W}_0=(W_{1,0},W_{2,0},..,W_{m,0})$. $t$=0.

2) Find a solution

$$x_{t+1} = \underset{\boldsymbol{x}}{\text{argmin}} \left\{ F(\boldsymbol{x},\boldsymbol{w}_t,\boldsymbol{W}_t) = f(\boldsymbol{x}) + \sum_r I(w_{r,t},W_{r,t}) h_r(\boldsymbol{x}) \mid \boldsymbol{x} \in N(\boldsymbol{x}_t) \right\}.$$

3) $\boldsymbol{w}_{t+1}= \boldsymbol{w}_t - \alpha_w \boldsymbol{w}_t + \lambda_w h_r(\boldsymbol{x}_{t+1})$. $\boldsymbol{W}_{t+1}= \boldsymbol{W}_t - \alpha_W \boldsymbol{W}_t + \lambda_W h_r(\boldsymbol{x}_{t+1})$.

4) If $\boldsymbol{x}_{t+1}$ satisfies the terminating condition then halt, else $t$=$t$+1 and goto 2).

In LAGLS, $w_{r,t}$ is the short term memory of the constraint $C_r$ at step $t$ and $W_{r,t}$ is the corresponding value for long term memory. The Lagrangian function is extended to $F(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{W})$ and $\alpha_w$, $\lambda_w$ are decay factor and cost coefficient of short term memory, and $\alpha_W$, $\lambda_W$ are those of long term memory. As the names "short" and "long" indicate, $\alpha_w > \alpha_W$, i.e., the short term memories forget the distant past, while long term memories can accumulate the information stored within a long time interval.

In the above definition, function $I(w_{r,t}, W_{r,t})$ plays an important role. This function integrates the long and short term memories. There are several possible ways of defining the integration function. Among them, in this paper, we experimentally examine the effectiveness of the following:

$$I_1(w_{r,t}, W_{r,t}) = w_{r,t} \times W_{r,t}, \tag{1}$$

$$I_2(w_{r,t}, W_{r,t}) = w_{r,t} + W_{r,t}. \tag{2}$$

From here onwards, we refer to (1) and (2) as "$w \times W$" and "$w+W$" respectively. The main objective of them is to prevent the search from stopping at a non-feasible solution. Suppose that the current solution $\boldsymbol{x}_t$ is not a feasible solution where constraint $C_r$ is violated. Then the values of $w_r$ are increased according to 3) (in both of the LAG and LAGLS), because $h_r(\boldsymbol{x})$ has a positive value. Therefore, according to 2), the solutions in which $C_r$ is violated cannot be selected as $\boldsymbol{x}_{t+1}$. Hence the search does not stop at $\boldsymbol{x}_t$ (i.e., $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$). This is the intuitive explanation why Lagrangian method does not stop at a non-feasible solution. The short term memory not only remembers $\boldsymbol{x}_t$, but also remembers solutions visited in the recent past. This makes the possibility of revisiting recently visited solutions small. Hence the search proceeds forward until it reaches a feasible solution. However, using only short term memories cannot prevent the search from being trapped by a long limit cycle. But this can be prevented by using long term memories. Therefore, the role of integration function $I$ is to add these powers of long and short term memories to the Lagrangian method.

The long term memory may also increase $F(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{W})$ for solution $\boldsymbol{x}$ which is not visited yet, i.e. it may include incorrect information in the sense of "already visited solutions". Hence using too small $\alpha_W$ indicates use of wrong information for a long time. There may be other ways of defining the integration function $I$, e.g. $\max\{\boldsymbol{w}, \boldsymbol{W}\}$ and $\min\{\boldsymbol{w}, \boldsymbol{W}\}$. However, $\max\{\boldsymbol{w}, \boldsymbol{W}\}$ ($\min\{\boldsymbol{w}, \boldsymbol{W}\}$) cannot be used, because $\boldsymbol{w}$ does not contribute when $\boldsymbol{w} < \boldsymbol{W}$ ($\boldsymbol{w} > \boldsymbol{W}$), and vice versa. More complex combination of "×" and "+" can also be used, however, in this paper we discuss only the basic ones shown in 1) and 2).

## 3 Experiments

### 3.1 Software-MCU Assignment Problem (SMAP)

To show the effectiveness of the Lagrangian method, we apply it to the Software-MCU assignment problem. This is an optimization problem of finding a minimum cost Software-MCU (micro controller unit) assignment as shown in Fig.1. The problem can be described as follows:

1) A set of software modules $S=\{s_i \mid i=1, 2,\ldots,n_S\}$. Each software module $s_i$ consumes computation power comp($i$) and memory mem($i$). For each pair of software modules $s_i$ and $s_l$ there is a communication traffic comm($i$, $l$).
2) A computer network which consists of a set of mutually connected communication nodes $N=\{n_j \mid j=1, 2,\ldots,n_N\}$. Each node $n_j$ has a communication cost $cost_{comm,j}=\mathrm{cost}_{comm,j}(c)$, where $c$ is the total amount of communication traffic passing through the node $n_j$. Node $n_j$ also has a communication capacity COMM($j$).
3) A set of MCUs $M=\{n_k \mid k=1, 2,\ldots,n_M\}$. Each MCU is attached to a node in $N$. MCU $m_k$ has a computation cost function $cost_{comp,k}=\mathrm{cost}_{comp,k}(p)$, where $p$ is the total amount of computation consumption of the software modules assigned to the MCU. MCU $m_k$ has a computational capacity COMP($k$). For the memory, MCU $m_k$ has a cost function $cost_{mem,k}=\mathrm{cost}_{mem,k}(q)$ and a capacity MEM($k$).

Software-MCU assignment problem is to find a minimum cost assignment $\boldsymbol{x}$ of the software modules on MCUs, where $x_k$ is a MCU to which software module $s_k$ is assigned.

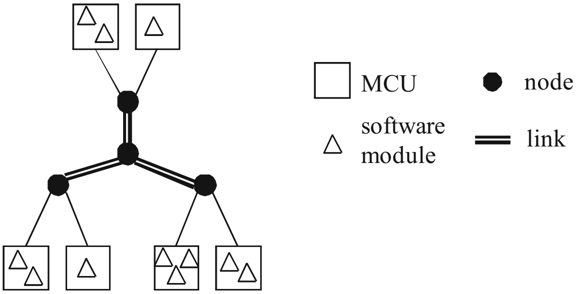


**Fig. 1.** An illustration for Software-MCU assignment problem (SMAP)

(Software-MCU assignment problem (SMAP))

$$\begin{array}{|l} \text{minimize}\, f(\boldsymbol{x}) = \sum\limits_k cost_{comp,k} + \sum\limits_k cost_{mem,k} + \sum\limits_j cost_{comm,j} \\ \text{subject to} \quad \sum\limits_{i|x_i=k} \mathrm{comp}(i) \le \mathrm{COMP}(k), \quad k=1,2,\ldots,n_M, \\ \qquad\qquad\quad \sum\limits_{i|x_i=k} \mathrm{mem}(i) \le \mathrm{MEM}(k), \quad k=1,2,\ldots,n_M, \\ \qquad\qquad\quad \sum\limits_{(i,l)\in\xi(j)} \mathrm{comm}(i,l) \le \mathrm{COMM}(j), \quad j=1,2,\ldots,n_N, \\ \qquad\qquad\quad x_i \in \{1,\ldots,n_M\}, \quad i=1,2,\ldots,n, \end{array}$$

where $\xi(j)$ is a set of pairs of software modules ($i$,$l$) such that the communication traffic between MCU $x_i$ and MCU $x_l$ passes through node $n_j$.

(Software-MCU assignment problem (unconstrained))

$$\left| \begin{aligned} &\text{minimize } \mathrm{g}(\boldsymbol{x}) = f(x) + \sum_k P\left( \sum_{i|x_i=k} \mathrm{comp}(i),\ \mathrm{COMP}(k) \right) \\ &\quad + \sum_k P\left( \sum_{i|x_i=k} \mathrm{mem}(i),\ \mathrm{MEM}(k) \right) + \sum_j P\left( \sum_{(i,l)\in\xi(j)} \mathrm{comm}(i,l),\ \mathrm{COMM}(j) \right) \end{aligned} \right.$$

We call $g(\boldsymbol{x})$ the "augmented objective function".

## 3.2 Lagrangian Method for the Software-MCU Assignment Problem

The Lagrangian method is implemented as follows to solve the Software-MCU assignment problem:

```
(LAGLS for SMAP)
  1) Make an initial assignment, $\boldsymbol{x}_0$ ,by any constructive method.
     $\boldsymbol{w}_0=(w_{1,0},w_{2,0},..,w_{m,0})$. $\boldsymbol{W}_0=(W_{1,0},W_{2,0},..,W_{m,0})$. $t=0$.
  2) Find a solution
     $\boldsymbol{x}_{t+1} = \underset{x}{\mathrm{argmin}}\{\sum_r I(w_{r,t},W_{r,t})\times h_r(\boldsymbol{x}) \mid \boldsymbol{x}\in N(\boldsymbol{x}_t)\}$
     where, $h_r(\boldsymbol{x}) = \begin{cases} cost_r & \text{if constraint } C_r \text{ is satisfied by } \boldsymbol{x}, \\ P(\sum consumption_r, capacity_r) & \text{otherwise.} \end{cases}$
  3) $w_{t+1} = w_t - \alpha_w w_t + \lambda_w \frac{h_r(\boldsymbol{x}_{t+1})}{\max_{r'}\{h_{r'}(\boldsymbol{x}_{t+1})\}}$,
     $W_{t+1} = W_t - \alpha_W W_t + \lambda_W \frac{h_r(\boldsymbol{x}_{t+1})}{\max_{r'}\{h_{r'}(\boldsymbol{x}_{t+1})\}}$.
  4) If $\boldsymbol{x}_{t+1}$ satisfies the terminating condition then halt, else $t$=$t$+1 and goto 2),
```

For simplicity, in 2), $r$ represents all types of constraints, i.e., *comp*, *mem*, and *comm*. "$cost_r$", "$capacity_r$" and "$consumption_r$", represent the cost, capacity, and consumption of $r$ respectively. In our experiments, each $w_{i,0}$, and $W_{i,0}$ are set to 0.1. As a constructive method for the initial solution, randomly selected software modules are selected one by one, and assigned to the best MCUs iteratively. For cost functions, quadratic functions are used with maximum value around 1.0. For the penalty function, $P(a,b)$, the following is used:

$$P(a,b) = \begin{cases} L(a/b)^2 & \text{if } a > b, \\ 0 & \text{otherwise.} \end{cases}$$

where $L$ is a large value, and we use $L$=50. The neighborhood $N(\boldsymbol{x})$ is defined to be a set of solutions which can be obtained by moving one software module or swapping two software modules.

### 3.3 Experimental Results of Solving SMAP via LAGLS for SMAP

In the experiments, we generate 6 problems of the SMAP. Each of them has 36 software modules, 12 MCUs and 6 nodes. Table 1 shows the results. Each row corresponds to a problem and each column corresponds to a way of integrating the long and short term memories. In the Table 1, "$w(\alpha$=0.03)" indicates the ordinary LAG, where 0.03 is the best value of $\alpha$ for almost all SMAP problems by LAG. Further, for "$w(\alpha$=0.03)" we used $\lambda$=1.0. For "$w\times W$", we used $\alpha_w$=0.03, $\lambda_w$=1.0, $\alpha_W$=0.0002 and $\lambda_W$=0.1, and for "$w+W$" $\lambda_W$=0.01 and other values are kept unchanged from "$w\times W$". After deciding the decay factor and cost coefficient for long term memory in the above integration methods, for the sake of completion, we decided to do an experiment using only long term memory. We did that experiment by replacing $\alpha$ and $\lambda$ in ordinary LAG by $\alpha_W$ and $\lambda_W$ of "$w+W$". The results are included in the Table 1 under $W(\alpha$=0.0002). All these are best values obtained from the preliminary experiments. For each problem and for each way of integration, we solved the problem 50 times by changing the initial solutions. The number of steps in each search is limited to 1000. In Table 1, each cell has 2 values, e.g. "665.7 / 29". The second value indicates that within 50 searches, a feasible solution can be found 29 times while the first value indicates the average number of steps required for each search. From this table we can see that "$w\times W$" is able to find the highest number of feasible solutions in all 6 problems. Furthermore, the number of steps required in "$w\times W$" is the least for 5 problems. Therefore, it is apparent that "$w\times W$" is the best for all problems, while "$w+W$" is worse than "$w(\alpha$=0.03)" for some problems. We think that the ability of escaping from local minima depends on the ratio of $\{I(w_r, W_r)|\ C_r$ is satisfied$\}$ and $\{I(w_r, W_r)|\ C_r$ is violated$\}$, and "$w\times W$" can make this ratio higher than "$w+W$" does. One search (=1000 steps) consumes around 50 sec on Intel 2.4GHz Quad CPU using only one core.

Fig.2 . shows how the value of the objective function, $f(\boldsymbol{x})$ ,changes with time. (a) is for "$w\times W$", and (b) is for "$w(\alpha$=0.03)". In (b), we can see that an oscillation occurs because of being trapped by limit cycles.

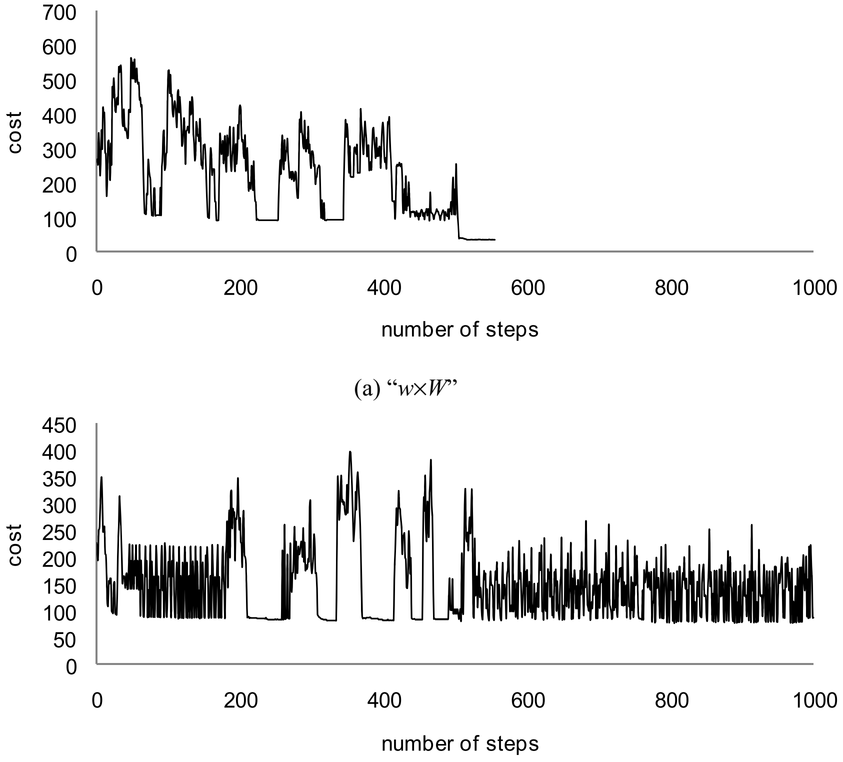
### 3.4 Using Another Heuristic Method for SMAP

In the experiment, we compared the LAGLS with the guided local search (GLS)[3]. The GLS and its derivatives are currently well used for optimization problems. The following is a description of the simple GLS.

```
(GLS)
  1) Make an initial solution $\boldsymbol{x}_0$.
     $\boldsymbol{P}=(p_{1,0}, p_{2,0},.., p_{m,0},)$. t=0.
  2) Apply the simple local search to $\boldsymbol{x}_t$, until being trapped by local minimal,
     and obtain $\boldsymbol{x}_{t+1}$ as follows:
       $h \leftarrow g + \lambda \sum p_i I_i$
       $x_{t+1} \leftarrow localseach\left(x_t, h, N\right)$
```

**Table 1.** Several ways of integrating long and short term memories are compared

| Problem | $w(\alpha=0.03)$ | $W(\alpha=0.0002)$ | $w\times W$ | $w+W$ |
|---|---|---|---|---|
| 1 | 665.7 / 29 | 860.2 / 10 | 535.2 / 41 | 728.9 / 20 |
| 2 | 737.4 / 23 | 846.7 / 12 | 533.9 / 36 | 626.5 / 30 |
| 3 | 275.4 / 44 | 447.1 / 33 | 207.1 / 48 | 217.7 / 47 |
| 4 | 368.7 / 39 | 419.6 / 35 | 364.0 / 41 | 439.5 / 37 |
| 5 | 338.6 / 42 | 422.5 / 35 | 340.7 / 43 | 421.1 / 42 |
| 6 | 821.8 / 18 | 859.8 / 12 | 794.1 / 20 | 878.4 / 14 |



(a) "$w\times W$"

(b) "$w(\alpha=0.03)$"

**Fig. 2.** Graphs to show how the values of objective function $f(\boldsymbol{x})$ changes with time

```
3) Choose a feature and penalize it as follows:
    for $i \leftarrow 1$ until $m$ do
        $util_i \leftarrow I_i(x_{t+1})c_i \big/ ((1+p_i))$;
    for each $i$ such that $util_i$ is maximum do
        $p_i \leftarrow p_i + 1$;
4) If $\boldsymbol{x}_{t+1}$ satisfies the terminating condition then halt, else $t$=$t$+1 and go to 2),
```

In this description, $m$ is the number of features, and $p_i$ and $I_i$ are penalty and the occurrence of the feature $i$, respectively. The neighborhood is represented by $N$, and it is similar to that of Lagrangian method. In 2), the function $h$ is the augmented objective function, while $g$ is the original one. We use only the violations of constraints as features. We applied the GLS to the problems in Table 1. However the GLS cannot find any feasible solutions for any problems. We confirmed that the GLS cannot escape from a local optimum. Even if we can resolve this situation by changing the definitions of the neighborhood or features, this results show at least the Lagrangian method is easier to design.

## 4 Conclusions

We propose a method to solve optimization problems which is based on the Lagrangian method. We also propose an integration method for short and long term memories. Experimental results show that these proposals are effective in finding feasible solutions. Currently we do not pay much attention to getting the true optimum solutions but only pay attention to getting feasible solutions. Experiments are done by solving hard problems for which even feasible solutions are difficult to find. For the future work we want to study about finding an optimum solution. From our previous work [6], we consider that the Lagrangian method also works well for this purpose.

## References

1. Glover, F.: Tabu Search Part I. ORSA Journal on Computing 1(3), 109–206 (1989)
2. Glover, F.: Tabu Search Part II. ORSA Journal on Computing 2(1), 4–32 (1990)
3. Voudouris, C.: Guided Local Search for Combinatorial Optimization Problems. PhD Thesis, Department of Computer Science, University of Essex, Colchester, UK (1997)
4. Nagamatu, M., Yanaru, T.: Solving SAT by Lagrange Programming Neural Network with Long and Short Term Memories. In: Kawaguchi, E., et al. (eds.) Information Modeling and Knowledge Bases XI, pp. 289–301. IOS Press, Amsterdam (2000)
5. Nagamatu, M.: Controlling the Diversive and Specific Exploration in Solving Disjunctive Linear Constraint Satisfaction Problem. International Journal of Biomedical Soft Computing and Human Sciences 13(2), 35–41 (2008)
6. Nakano, T., Nagamatu, M.: A Continuous Valued Neural Network with a New Evaluation Function of Degree of Unsatisfaction for Solving CSP. IEICE Transactions on Information and Systems E89-D(4), 1555–1562 (2006)

# Overlap-Based Similarity Metrics for Motif Search in DNA Sequences

Hai Thanh Do and Dianhui Wang

Department of Computer Science and Computer Engineering
La Trobe University, Melbourne, Victoria, 3086, Australia
dh.wang@latrobe.edu.au

**Abstract.** Motifs refer to a collection of transcription factor binding sites (TFBSs) which are located at promoters of genes. Discovery of motifs is critical to further understanding the mechanism of gene regulation. Computational approaches addressing this challenging problem have demonstrated good potential. However, the existing motif search approaches have some limits to deal with remarkably under-presentation of binding sites in biological datasets, resulting in considerably high false-positive rate in prediction. We resolve the task as an imbalanced biological data classification problem and our technical contributions in this paper include the following aspects: (i) propose a novel similarity metrics for comparing DNA subsequences based on overlap range of nucleotides in DNA sequences; and (ii) introduce a new sampling method which combines both over- and under-sampling techniques. The effectiveness of our proposed similarity metrics and sampling approach is demonstrated by two benchmark datasets and three classification techniques – Neural Networks (NN), Support Vector Machine (SVM), and Learning Vector Quantization (LVQ1). Empirical studies show that the classifier LVQ1 integrated with the proposed similarity metrics performs slightly better other approaches on the test datasets.

## 1 Introduction

Transcription plays a principal role in the central dogma of molecular biology or gene expression. It transcribes genetic information (genes) embedded in DNA sequences into messenger RNA segments from which corresponding functional proteins are produced. In order to be accomplished, the transcription requires a sequence-specific binding of multiple proteins or Transcription Factors (TFs) to short stretches of DNAs (Transcription Factor Binding Sites - TFBSs or binding sites for short) in promoter regions of being-regulated genes. It is biologically verified that this DNA-protein interaction underlies the success and the level of gene expression [1]. The identification of these TFs-targeted sites, therefore, is crucial for a complete understanding of gene regulatory complex networks [2]. Due to being biologically functional, TFBSs of a certain TF are likely conserved across evolutionary lineages, resulting in common patterns or *motifs* of them.

Motif search [3,2] can be formulated as a problem of finding putative TFBSs associated with a TF in the promoter regions of interest based on given experimentally TFBSs identified using computational techniques. As such, two

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 465–474, 2009.

challenges must be taken. First, TFBSs are degenerate, i.e., variations are allowable at a position within sites due to effects of species selection. Second, the comparative shortness of TFBSs (5-20 base pairs) together with the long length of promoter regions (hundreds base pairs in plants or thousands in human) often brings about high probabilities of finding similar sequences by chance [4].

Two issues are related to the problem of motif search. On one hand, research efforts have been made on improving the representation of motifs, from a Consensus Sequence [5] just listing nucleotides permissible at each position, to a more descriptive Position Frequency Matrix [5] representing occurring probabilities of each nucleotide at each position of aligned binding sites. These two representation methods assume the independence of site positions while recent laboratory experiments have shown that there exists inter-position dependence within sites [2]. Thus, current research work has taken this dependency into consideration [6,7,8]; yet the system performance could not be improved universally. On the other hand, attempts has been made to reduce the higher false-positive rate in binding sites prediction. Some approaches enhanced the differentiation power of scoring schemas used to measure the degree to which a new sequence matches the given motifs [3,2]; some took advantage of the blossom of sequenced genome data by incorporating prior knowledge, for example, phylogenetic footprinting [9,10]. Unfortunately, a satisfying prediction accuracy still remains challenging [3].

The nature of motif search drives us to resolve the task as an imbalanced biological data classification problem, and consequently, the following two issues must be considered: (i) the similarity measure used in classifiers design and (ii) the remarkable imbalance of data that needs classifying. For the first issue, the Hamming distance or Euclidian norm is normally employed. However, it is obviously unsuitable for motif search tasks because two almost-overlap subsequences within the binding sites region appear totally dissimilar. For the second issue, sampling techniques (over-sampling [11,12] and under-sampling [13]) are often used. The combination of the two sampling techniques is necessary due to two reasons: (i) the size of the positive set is clearly under-presented in comparison with that of the negative set in the training data; and (ii) the greater number of negative examples do not bring a higher classification performance [11].

This paper aims to tackle the above two issues for further biological data mining practice. In particular, we propose a new similarity metrics based on overlapping for comparing nucleotide subsequences. Accordingly, a sampling method is presented. The effectiveness of both the proposed similarity metrics and the sampling approach is demonstrated by two benchmark datasets and three well-known classifiers, namely Neural Networks, Support Vector Machine, and Learning Vector Quantization.

The rest of this paper is organized as follows: the proposed similarity metrics, sampling technique, and an encoding method are presented in Section 2. Experimental results with comparisons are reported in Section 3. Finally, Section 4 concludes the paper.

## 2 Method

### 2.1 Encoding Method

Subsequences of nucleotides with length $k$, namely kmers, are commonly encoded by a binary coder [14]. Recently, [15] proposed a binary matrix encoding approach that concides with the position frequency matrix (PFM) representation for motifs. However, these encoding methods does not suit well for building classifiers. In this paper, we convert DNA sequences into numerical feature vectors by counting consecutive joint frequency.

Let a kmer be $T_1T_2...T_k$, where $T_j \in \sum = \{A, C, G, T\}, j = 1, 2, ..., k$. We construct an encoding matrix $M = [m_{ij}]_{4\times4}$ as $m_{ij} = \text{freq}(V_i, V_j)$, where $(V_1, V_2, V_3, V_4) = (A, C, G, T)$ and $\text{freq}(V_i, V_j)$ is the occurrence frequency of the ordered pair $(V_iV_j)$ in the working dataset. Elements of the matrix are then normalized by $m_{ij} = (m_{ij} - Min)/(Max - Min)$, where $Min$ and $Max$ are the minimum and maximum elements of the matrix $M$.

The encoded kmers are now given by $e(\text{kmer}) = [e_i]_{1\times(k-1)}, e_i = m_{jl}$ if $T_i = V_j$ and $T_{i+1} = V_l$. If we let $N$ be the total number of kmers, encoding in such a way will shrink the input space from $N \times k \times 4$ to $N \times (k-1)$, and as a result, decreasing remarkably the required processing time.

### 2.2 Overlap-Based Similarity Metrics

As discussed in [16], motif search performance depends on overlapping concept. A predicted site which overlaps over some positions with a known site should also be counted as a true positive. Such a predicted site, however, may be treated as a negative in the sense of the Hamming distance and Euclidian norm. This phenomena brings us a motivation to develop the overlap-based similarity metrics (OSIM), in which the best alignment of two DNA subsequences is targeted.

Let $E_1$ and $E_2$ be two encoded kmers, i.e., $E_i = e^i_1e^i_2...e^i_{k-1}$, where $e^i_j \in R$, $i = 1, 2$, and $j = 1, 2, ..., k-1$. In order to be evaluated as true positive, a kmer must overlap with a known site over at least a specified number of positions. We represent this number as $\theta$, where $1 \leq \theta \leq (k-1)$, and call it the overlapping parameter. The distance between $E_1$ and $E_2$, $d(E_1, E_2)$, is formulated as the minimum distance obtained in every possible move $t$ of $E_2$ with respect to $E_1$, where $\theta \leq t \leq (k-1)$. For each move $t$, we shift $E_2$ to the left $(k-1-t)$ positions with respect to $E_1$ and compute the left-aligned distance $(d^t_L(E_1E_2))$. Similarly, we compute the right-aligned distance $(d^t_R(E_1, E_2))$ by shifting $E_2$ to the right the same number of positions. The smaller value between $d^t_L(E_1, E_2)$ and $d^t_R(E_1, E_2)$ is recorded as the distance between $E_1$ and $E_2$ in this move, $d^t(E_1, E_2)$. Finally, $d(E_1, E_2)$ is decided to be the smallest value among $(k-\theta)$ values of $d^t(E_1, E_2)$.

$$d(E_1, E_2) = \min_{\forall t} \{d^t(E_1, E_2)\}, t = \theta, \theta + 1, ..., k-1, \tag{1}$$

where $d^t(E_1, E_2)$ is given by

$$d^t(E_1, E_2) = \min\{d^t_L(E_1, E_2), d^t_R(E_1, E_2)\}, \tag{2}$$

$$d^t_L(E_1, E_2) = \sum_{i=1}^{t} |e^{(1)}_{i+k-t-1} - e^{(2)}_i| + \delta_t, \tag{3}$$

$$d^t_R(E_1, E_2) = \sum_{i=1}^{t} |e^{(2)}_{i+k-t-1} - e^{(1)}_i| + \delta_t, \tag{4}$$

where $\delta_t$ is the penalty of the position shifting and variable with respect to $t$; the higher $t$, the smaller $\delta_t$. The selection of $\delta_t$ is arbitrary and should be dataset-dependent. We implement OSIM with $\delta_t = (k-t-1)/(k-1)$ in this work. When $t = k-1$, the two kmers completely overlap each other and OSIM becomes the widely-used Hamming distance.

OSIM has two advantages. First, it can work on both data space and feature space. Second, the use of OSIM avoids the situation in which two kmers close to each other on a sequence appear completely dissimilar. However, when known binding sites are short or $k$ is small, the probability of accidently finding a similar sequence is high, and so, applying OSIM will add more false positives in prediction. Fortunately, we can manage the value of $k$ and binding sites are fluctuate in length.

### 2.3 Training Examples Generation

We use both under- and over-sampling techniques to generate training data for classifiers. In a recent work [13], the authors proposed an over-sampling technique called SMOTE and showed good classification results. However, SMOTE is inappropriate for the motif search task we are tackling. We work on kmers while new positive examples generated by SMOTE are not kmers. Thus, we present here an over-sampling method based on overlapping concept. Also, a new under-sampling technique is introduced since the common random under-sampling approaches are highly subjective to the loss of information.

Each original imbalanced training dataset is composed of kmers, in which annotated binding sites (or positive examples) contribute a very small portion. The positive set is over-sampled as the followings. For each known site $bs_i$, we find kmers overlapping with $bs_i$ over (at least) $p$ positions. Then, we add them into the positive set and simultaneously remove them from the kmer set. $p$ should be set as the same as the overlapping parameter $\theta$.

After over-sampling, we under-sample negative examples by selecting non-functional kmers under the sense of OSIM. For each kmer in the kmer set, we construct a distance vector whose elements are distances (calculated using OSIM) from the kmer to all positive examples. These distance vectors are then sorted in ascending order based on a so-called minimum-criteria, that is, if $min(v_i) \geq min(v_j)$ then $v_i \geq v_j$, where $v_i$ and $v_j$ are two distance vectors. After that, $N$ kmers associated with the top $N$ distance vectors are selected for negative examples. $N$ equals the multiplication of the imbalance factor (set at

1) and the number of positive examples. Investigating the effect of imbalance factor on classifiers is beyond the scope of the paper.

### 2.4 Learner Models

Three supervised classifiers are investigated for a systematic comparison study. Backpropagation neural network (BPNN) [17] is a multi-layer neural network that can learn the structure of data through representatives using the backpropagation learning algorithm. Knowledge learned is stored in weighted connections from the input layer through hidden layers and to the output layer. Support vector machine (SVM) [17] aims to construct an optimal separating hyperplane with maximal margin for classifying data points. Learning vector quantization 1 (LVQ1) [18] attempts to build up vector quantization from training data. In addition, to evaluate the merits of our proposed similarity metrics, we build an OSIM-based version of LVQ1 by replacing the Euclidian norm used to specify the winning neuron with our proposed similarity metrics.

### 2.5 Performance Evaluation

Predictions of classifiers in this paper are evaluated in terms of precision ($Pre$), recall ($Re$), and $F$-measure ($F$) [19]. Overlapping is also deployed to determine true positives (TP), false positives (FP), and false negatives (FN) as represented in [16]. We define that a predicted site overlaps with a known site if it overlaps by at least half of its length with the known site.

Predicted binding sites of a classifier are kmers recognized as positive. For each classifier and dataset, let $sTP$, $sFP$, and $sFN$ be the number of predicted sites overlapped by known sites, the number of predicted sites not overlapped by known sites, and the number of known sites not overlapped by predicted sites, respectively [16]. Precision, recall, and $F$-measure are then given by: $Pre = sTP/(sTP+sFP)$, $Re = sTP/(sTP+sFN)$, and $F = (2*Pre*Re)/(Pre+Re)$.

## 3 Empirical Results

### 3.1 Data Sets

The benchmark dataset used in this paper includes 8 real sets of human promoter sequences and 8 synthetic ones, corresponding to 8 various transcription factors (Table 1). Human promoters with known sites were collected from the well-studied ABS database [20]. Artificial sequences with equal distribution of nucleotides were generated by using sequence generation tool of ABS. In each synthetic set, one site motif of the corresponding TF is implanted with probability of 0.75. All sequences in the 16 sets are 500 base-pairs in length.

The dataset has three characteristics: (i) the data imbalance in each set is high; (ii) there are significant variations of the data size (from 8 to 42, real sets) and the length of binding sites (from 5 to 46), and (iii) binding sites in the real

**Table 1.** DataSet Statistics (*Transcrption Factor, Number of Promoter (Artificial) Sequences, Number of Annotated Binding Sites, Min Binding Site Length, Max Binding Site Length*)

(a) Real Data

| TF | No. | No. | Min | Max |
|---|---|---|---|---|
| | PSs | ABSs | BSL | BSL |
| CEBP | 17 | 37 | 6 | 32 |
| CREB | 9 | 19 | 5 | 22 |
| E2F1 | 8 | 16 | 8 | 15 |
| HNF1 | 12 | 27 | 11 | 29 |
| MEF2 | 6 | 17 | 7 | 14 |
| SP1 | 42 | 89 | 6 | 46 |
| SRF | 14 | 36 | 9 | 22 |
| TBP | 39 | 95 | 5 | 24 |

(b) Artificial Data

| TF | No. | No. | Min | Max |
|---|---|---|---|---|
| | ASs | ABSs | BSL | BSL |
| CEBP | 10 | 6 | 11 | 32 |
| CREB | 10 | 6 | 8 | 22 |
| E2F1 | 10 | 7 | 8 | 15 |
| HNF1 | 10 | 5 | 13 | 25 |
| MEF2 | 10 | 5 | 9 | 14 |
| SP1 | 10 | 8 | 6 | 18 |
| SRF | 10 | 8 | 10 | 22 |
| TBP | 10 | 8 | 6 | 7 |

sets are very unconserved while those in the artificial ones are more conserved. With the above characteristics of the benchmark dataset, we consider most likely scenarios for assessing our work.

### 3.2 Experiment Setup

The experiments were conducted on computer with a Pentium 4, 3.00GHz CPU and 1 GB memory. The software was implemented in Matlab based on two toolboxes: Neural Network Matlab Toolbox and Bioinformatics Matlab Toolbox.

For each dataset, five-fold standard cross validation approach was adopted. With each fold, the training data was segmented into kmers from which two training data types for classifiers were generated. The first type, *resampled data*, was produced by using the proposed sampling technique. The other one, *non-resampled data*, was built up as follows. We employed CLUSTALW [21] to align annotated binding sites (if needed). Then, we manually extracted $k$-base-pair subsequences in the most conserved portion of the alignment and used them as positive examples. Negative examples were kmers left after positive ones were taken from the kmer list.

Three experiment sets were set up. First, to evaluate the recognizability of our proposed sampling method, we sequentially trained BPNN and SVM with the two training data types. We did not compare the proposed sampling approach with existed ones because they generate artificial positive examples that are not kmers (such as SMOTE).

Second, the feasibility of OSIM was assessed through predictions of LVQ1 and OSIM-LVQ1 after the two classifiers had been trained with resampled data. The overlapping parameter $\theta$ was set as half of kmers length ($k/2$). Classifiers in all experiments were executed 5 times corresponding to 5 folds, and average performance of each classifier on 5 testing data portions was reported.

Third, to provide additional support for our proposed OSIM and sampling approach, we used P-Match [22] to search for binding sites on 8 real datasets. We intended to test the recall-ability of P-Match, so all matrices in TRANSFAC [23] (not only high quality ones) were used. The cut off parameter was set to minimize false positives. Results obtained from P-Match were compared with those from the OSIM-LVQ1.

### 3.3 Results and Discussion

Performance metrics of the SVM trained with the non-resampled data are all zeros, while that of the SVM trained with resampled data are shown in Table 2. Table 3 presents classification capability of BPNN after sequentially trained with resampled and non-resampled data. It is clearly seen from those tables that SVM and BPNN can not perform the classification on the 16 datasets without re-balancing the training data. With our proposed sampling technique, they are able to recognize almost all binding sites annotated in both real and artificial test data.

**Table 2.** Classification Results of SVM with Sampling Technique (*linear kernel, sequential minimal optimization)*

(a) Artificial Data

| **TF** | **Pre** | **Re** | **F** |
|---|---|---|---|
| CEBP | 0.03 | 1.00 | 0.07 |
| CREB | 0.03 | 1.00 | 0.06 |
| E2F1 | 0.03 | 1.00 | 0.07 |
| HNF1 | 0.02 | 0.80 | 0.03 |
| MEF2 | 0.01 | 0.80 | 0.02 |
| SP1 | 0.09 | 0.80 | 0.15 |
| SRF | 0.02 | 0.80 | 0.05 |
| TBP | 0.04 | 1.00 | 0.07 |

(b) Real Data

| **TF** | **Pre** | **Re** | **F** |
|---|---|---|---|
| CEBP | 0.08 | 1.00 | 0.12 |
| CREB | 0.09 | 1.00 | 0.11 |
| E2F1 | 0.04 | 1.00 | 0.07 |
| HNF1 | 0.06 | 1.00 | 0.12 |
| MEF2 | 0.03 | 1.00 | 0.05 |
| SP1 | 0.07 | 1.00 | 0.14 |
| SRF | 0.07 | 1.00 | 0.13 |
| TBP | 0.02 | 1.00 | 0.03 |

**Table 3.** Classification Results of BPNN: (a) and (b): with Sampling Technique (*learning rate = 0.01, number of epochs = 500, number of hidden neurons = 5*); (c) and (d): without Sampling Technique (*learning rate = 0.01, number of epochs = 500, number of hidden neurons = 5*)

(a) Artificial Data

| **TF** | **Pre** | **Re** | **F** |
|---|---|---|---|
| CEBP | 0.03 | 0.80 | 0.06 |
| CREB | 0.03 | 1.00 | 0.05 |
| E2F1 | 0.02 | 0.90 | 0.04 |
| HNF1 | 0.02 | 0.80 | 0.04 |
| MEF2 | 0.01 | 0.60 | 0.02 |
| SP1 | 0.04 | 0.60 | 0.07 |
| SRF | 0.02 | 0.60 | 0.03 |
| TBP | 0.01 | 1.00 | 0.02 |

(b) Real Data

| **TF** | **Pre** | **Re** | **F** |
|---|---|---|---|
| CEBP | 0.08 | 1.00 | 0.12 |
| CREB | 0.02 | 0.90 | 0.04 |
| E2F1 | 0.04 | 1.00 | 0.07 |
| HNF1 | 0.05 | 0.80 | 0.09 |
| MEF2 | 0.02 | 0.87 | 0.04 |
| SP1 | 0.08 | 0.85 | 0.15 |
| SRF | 0.09 | 1.00 | 0.15 |
| TBP | 0.02 | 0.83 | 0.04 |

(c) Artificial Data

| **TF** | **Pre** | **Re** | **F** |
|---|---|---|---|
| CEBP | 0.01 | 0.23 | 0.02 |
| CREB | 0.01 | 0.43 | 0.02 |
| E2F1 | 0.00 | 0.10 | 0.00 |
| HNF1 | 0.02 | 0.20 | 0.03 |
| MEF2 | 0.00 | 0.00 | 0.00 |
| SP1 | 0.03 | 0.30 | 0.04 |
| SRF | 0.01 | 0.10 | 0.01 |
| TBP | 0.02 | 0.30 | 0.02 |

(d) Real Data

| **TF** | **Pre** | **Re** | **F** |
|---|---|---|---|
| CEBP | 0.02 | 0.26 | 0.02 |
| CREB | 0.01 | 0.10 | 0.01 |
| E2F1 | 0.02 | 0.20 | 0.01 |
| HNF1 | 0.02 | 0.14 | 0.03 |
| MEF2 | 0.03 | 0.30 | 0.04 |
| SP1 | 0.04 | 0.30 | 0.06 |
| SRF | 0.04 | 0.21 | 0.03 |
| TBP | 0.03 | 0.29 | 0.04 |

Table 4 reports the average performance of LVQ1 and OSIM-LVQ1. F-measure(s) of the four classifiers are also visualized in Figure 1 for comparison. As shown in Table 4, OSIM-LVQ1 performs better LVQ1 on 3 synthetic sets (CEBP, CREB, and SP1), and on 5 real ones (CEBP, CREB, MEF2, SP1, SRF). Most of real sets listed above are larger ones and known sites in those sets are long and highly unconserved. Meanwhile, E2F1, HNF1, and TBP, on which OSIM-LVQ1 perform slightly worse than LVQ1, are smaller sets (except TBP) and contain short and more conserved known binidng sites. Besides, we also see obviously from Figure 1 that OSIM-LVQ1 performs better than all of the other three classifiers on 4 real sets (CEBP, CREB, MEF2, SP1). Results reported on real sets are more significant than those on artificial ones. Predictions on synthetic data only demonstrate the independence of the proposed OSIM and sampling technique on the conservation of known sites.

**Table 4.** Classification Results of LVQ1 and OSIM-based LVQ1 with Sampling Technique on Artificial Data (A. Data) and Real Data (R. Data)(*number of epochs = 500, number of hidden neurons = 5*)

(a) A. Data-LVQ1

| TF | Pre | Re | F |
|---|---|---|---|
| CEBP | 0.03 | 1.00 | 0.06 |
| CREB | 0.04 | 1.00 | 0.08 |
| E2F1 | 0.03 | 1.00 | 0.06 |
| HNF1 | 0.03 | 1.00 | 0.05 |
| MEF2 | 0.02 | 1.00 | 0.04 |
| SP1 | 0.09 | 0.67 | 0.16 |
| SRF | 0.04 | 1.00 | 0.08 |
| TBP | 0.05 | 1.00 | 0.09 |

(b) R. Data-LVQ1

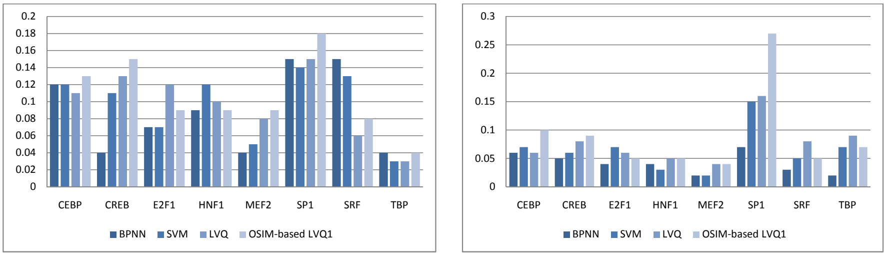
| TF | Pre | Re | F |
|---|---|---|---|
| CEBP | 0.05 | 1.00 | 0.11 |
| CREB | 0.07 | 1.00 | 0.13 |
| E2F1 | 0.06 | 1.00 | 0.12 |
| HNF1 | 0.05 | 1.00 | 0.10 |
| MEF2 | 0.04 | 1.00 | 0.08 |
| SP1 | 0.08 | 1.00 | 0.15 |
| SRF | 0.03 | 1.00 | 0.06 |
| TBP | 0.02 | 1.00 | 0.03 |

(c) A. Data-OSIM

| TF | Pre | Re | F |
|---|---|---|---|
| CEBP | 0.05 | 1.00 | 0.10 |
| CREB | 0.04 | 1.00 | 0.09 |
| E2F1 | 0.02 | 1.00 | 0.05 |
| HNF1 | 0.03 | 1.00 | 0.05 |
| MEF2 | 0.02 | 1.00 | 0.04 |
| SP1 | 0.15 | 0.95 | 0.27 |
| SRF | 0.03 | 1.00 | 0.05 |
| TBP | 0.04 | 0.67 | 0.07 |

(d) R. Data-OSIM

| TF | Pre | Re | F |
|---|---|---|---|
| CEBP | 0.06 | 1.00 | 0.13 |
| CREB | 0.08 | 1.00 | 0.15 |
| E2F1 | 0.05 | 1.00 | 0.09 |
| HNF1 | 0.05 | 1.00 | 0.09 |
| MEF2 | 0.04 | 1.00 | 0.09 |
| SP1 | 0.09 | 1.00 | 0.18 |
| SRF | 0.04 | 1.00 | 0.08 |
| TBP | 0.01 | 1.00 | 0.04 |



(a) Real Data (b) Artificial Data

**Fig. 1.** Performance statistics comparing the F-measure of four classifiers

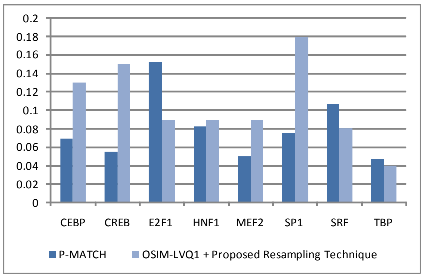


**Fig. 2.** Performance statistics comparing the F-measure of P-Match and OSIM-based LVQ1

**Table 5.** Recognition Results of P-Match on Real Datasets

| TF | No. Sites | Pre | Re | F |
|---|---|---|---|---|
| CEBP | 829 | 0.039 | 0.351 | 0.069 |
| CREB | 339 | 0.029 | 0.420 | 0.055 |
| E2F1 | 164 | 0.085 | 0.688 | 0.151 |
| HNF1 | 513 | 0.045 | 0.519 | 0.082 |
| MEF2 | 290 | 0.027 | 0.294 | 0.050 |
| SP1 | 1294 | 0.045 | 0.236 | 0.075 |
| SRF | 453 | 0.062 | 0.389 | 0.107 |
| TBP | 1526 | 0.025 | 0.400 | 0.047 |

Table 5 shows recognition results of P-Match on 8 real sets. It is obvious that recognizability of P-Match is not as good as OSIM-LVQ1, which can recall almost all annotated binding sites in the 8 real sets (as shown in Table 4). Visualization of F-measures in Figure 2 also presents that OSIM-LVQ1 performs better P-Match on most datasets.

Nevertheless, the precision reported in all experiments is still low. It has been studied that traditional classifiers are severely limited in accuracy on highly imbalanced data [24]. Biological data possesses many other difficulties not just the imbalance, with the human genome being even more challenging. If some pre- and pos-processing operations are taken, classification performance might be improved [25]. In this work, we only aim to demonstrate the workability and higher performance of OSIM over Euclidian norm so that OSIM can be exploited in further motif search practice. Achieving significant precision will be the goal of our further research.

## 4 Conclusion

In this work, we introduce a novel overlap-based metrics for measuring the similarity of nucleotide sub-sequences and a sampling technique, as a foundation for further motif searching practice. We demonstrate the effectiveness of them on both synthetic and genomic data using three traditional classifiers. The two most significant results presented are: (i) classifiers integrated with the proposed sampling technique have a higher recall rate; and (ii) the classifier LVQ1 embedded with the proposed similarity metrics works better on real datasets in which annotated binding sites are highly unconserved and long. However, the parameters significantly affect the performance of the similarity metrics. The penalty term, $\delta$, is set arbitrarily in this paper. Further research in this direction may address the followings: (i) building a more systematic setting model for $\delta$; and (ii) exploring some ways of using our proposed similarity metrics in classifiers design to achieve better prediction performance.

## References

1. Latchman, D.S.: Eukaryotic Transcription Factors. Elsevier, Amsterdam (2008)
2. Osada, R., Zaslavsky, E., Singh, M.: Comparative analysis of methods for representing and searching for transcription factor binding sites. Bioinformatics 20(18), 3516–3525 (2004)
3. Hannenhalli, S.: Eukaryotic transcription factor binding sites–modeling and integrative search methods. Bioinformatics 24(11), 1325–1331 (2008)
4. Sinha, A.U., Phatak, M., Bhatnagar, R., Jegga, A.G.: Identifying functional binding motifs of tumor protein p53 using support vector machines. In: Pro. The Sixth International Conference on Machine Learning and Applications (ICMLA 2007), Washington, DC, USA, pp. 506–511. IEEE Computer Society Press, Los Alamitos (2007)
5. Stormo, G.D.: DNA binding sites: representation and discovery. Bioinformatics 16(1), 16–23 (2000)
6. Chin, F., Leung, H.C.M.: DNA motif representation with nucleotide dependency. IEEE/ACM Trans. Comput. Biol. Bioinformatics 5(1), 110–119 (2008)
7. Zhou, Q., Liu, J.S.: Modeling within-motif dependence for transcription factor binding site predictions. Bioinformatics 20(6), 909–916 (2004)
8. Tomovic, A., Oakeley, E.J.: Position dependencies in transcription factor binding sites. Bioinformatics 23(8), 933–941 (2007)

9. Cliften, P.F.: 22 phylogenetic footprinting. In: Stansfield, I., Stark, M.J. (eds.) Yeast Gene Analysis, 2nd edn. Methods in Microbiology, vol. 36, pp. 551–576. Academic Press, London (2007)
10. Wasserman, W.W., Sandelin, A.: Applied bioinformatics for the identification of regulatory elements. Nat. Rev. Genet. 5(4), 276–287 (2004)
11. Gu, Q., Cai, Z., Zhu, L., Huang, B.: Data mining on imbalanced data sets. In: Proc. IEEE International Conference on Advanced Computer Theory and Engineering (ICACTE 2008), Cairo, Egypt, pp. 1020–1024 (2008)
12. Visa, S., Ralescu, A.: Issues in mining imbalanced data sets - a review paper. In: Pro. The Sixteen Midwest Artificial Intelligence and Cognitive Science Conference, Dayton, Ohio, USA, pp. 67–73 (2005)
13. Chawla, N.V., Bowyer, K.W., Kegelmeyer, P.W.: Smote: Synthetic minority over-sampling technique. J. Artif. Intell. Res (JAIR) 16, 321–357 (2002)
14. Liu, D., Xiong, X., DasGupta, B., Zhang, H.: Motif discoveries in unaligned molecular sequences using self-organizing neural networks. IEEE Trans. Neural Netw. 17(4), 919–928 (2006)
15. Wang, D.H., Lee, N.K.: MISCORE: mismatch-based matrix similarity scores for DNA motifs detection. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008, Part I. LNCS, vol. 5506, pp. 478–485. Springer, Heidelberg (2009)
16. Tompa, M., Li, N., Bailey, T.L., Church, G.M., Moor, B.D., Eskin, E., Favorov, A.V., Frith, M.C., Fu, Y., Kent, W.J., Makeev, V.J., Mironov, A.A., Noble, W.S., Pavesi, G., Pesole, G., Regnier, M., Simonis, N., Sinha, S., Thijs, G., van Helden, J., Vandenbogaert, M., Weng, Z., Workman, C., Ye, C., Zhu, Z.: Assessing computational tools for the discovery of transcription factor binding sites. Nat. Biotechnol. 23(1), 137–144 (2005)
17. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification, 2nd edn. Wiley-Interscience, Hoboken (2000)
18. Ham, F., Kostanic, I.: Principles of neurocomputing for science and engineering. McGraw-Hill Higher Education (2000)
19. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Cambridge University Press, Cambridge (2008)
20. Blanco, E., Farré, D., Albá, M., Messeguer, X., Guigó, R.: Abs: A database of annotated regulatory binding sites from orthologous promoters. Nucl. Acids Res. 34, 63–67 (2006)
21. Thompson, J., Higgins, D., Gibson, T.: Clustal w: improving the sensitivity of progressive multiple sequence alignments through sequence weighting, position specific gap penalties and weight matrix choice. Nucl. Acids Res. 22, 4673–4680 (1994)
22. Chekmenev, D.S., Haid, C., Kel, A.E.: P-match: Transcription factor binding site search by combining patterns and weight matrices. Nucl. Acids Res. 33, W432–W437 (2005)
23. Heinemeyer, T., Chen, X., Karas, H., Kel, A.E., Kel, O.V., Liebich, I., Meinhardt, T., Reuter, I., Schacherer, F., Wingender, E.: Expanding the transfac database towards an expert system of regulatory molecular mechanisms. Nucl. Acids Res. 27, 318–322 (1999)
24. Tang, Y., Zhang, Y.Q., Chawla, N., Krasser, S.: SVMs modeling for highly imbalanced classification. IEEE Trans. Syst. Man Cybern. Part B Cybern. 39(1), 281–288 (2009)
25. Robinson, M., Sun, Y., Boekhorst, R.T., Kaye, P., Adams, R., Davey, N., Rust, A.G.: Improving computational predictions of cis-regulatory binding sites. Pac. Symp. Biocomput. 11, 391–402 (2006)

# An Evolutionary Artificial Neural Network for Medical Pattern Classification

Shing Chiang Tan[1], Chee Peng Lim[2], Kay Sin Tan[3], and Jose C. Navarro[4]

[1] Faculty of Information Science & Technology, Multimedia University
Melaka Campus, Jalan Ayer Keroh Lama, Bukit Beruang, 75450 Melaka, Malaysia
sctan@mmu.edu.my
[2] School of Electrical & Electronic Engineering, University of Science Malaysia
Engineering Campus, 14300 Nibong Tebal, Penang, Malaysia
[3] Department of Medicine, Faculty of Medicine, University of Malaya,
50603 Kuala Lumpur, Malaysia
[4] University of Santo Tomas Hospital, Espana, Manila Philippines

**Abstract.** In this paper, a novel evolutionary artificial neural network based on the integration between Fuzzy ARTMAP (FAM) and a Hybrid Genetic Algorithm (HGA) is proposed for tackling medical pattern classification tasks. To assess the effectiveness of the proposed FAM-HGA model, the Ripley artificial data set is first used, and the results are compared with those of FAM-GA and FAM. A real medical data set comprising anonymous stroke patient records is then employed for further experimentation. The performance of FAM-HGA is assessed using three indicators; accuracy, sensitivity and specificity, and the results are compared with those of FAM-GA and FAM. Overall, FAM-HGA yields better classification performances than FAM-GA and FAM. The study reveals the potential of FAM-HGA as a computerized decision support tool for medical pattern classification tasks.

**Keywords:** Fuzzy ARTMAP, Hybrid Genetic Algorithms, Pattern Classification, Medical Decision Support.

## 1 Introduction

Medical prognostic and diagnostic problems are prime examples of decision making under uncertainty. Normally, these problems are characterized by more than one item of data. It is uncommon for a clinician to decide a disease based on a symptom or measurement without associating the symptom/measurement with others. Many outcomes may be drawn by referring to the same set of clinical data. On the other hand, different sets of data may lead to the same disease. This indicates that accurate prognosis/diagnosis of a disease is not an easy task as it involves the consideration of many factors that may cause complication and time delay therein (Yan et al., 2006). For example, heart diseases can be a causal reason to many syndromes. Apart from the heart, clinical symptoms as well as functional and pathologic manifestations of the heart can be related to other human organs. If an accurate diagnosis decision can be made, then an appropriate treatment can be arranged on time for heart disease

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 475–482, 2009.

patients. Therefore, it is potentially useful if automated computerized intelligent systems can be devised to improve the decision making process in the field of medical consultation and diagnosis.

Apart from data-driven analytical techniques such as decision and classification theory, an intelligent system can employ knowledge-based and evolutionary adaptation approaches to increase the capability of decision making. In the medical domain, these approaches have been applied by a number of researchers, e.g., Yan et al. (2006) and Abbass (2002). In this paper, an evolutionary artificial neural network (EANN), which is based on a fusion between Fuzzy ARTMAP (FAM) (Carpenter et al., 1992) and Hybrid Genetic Algorithm (HGA) (Baskar et al., 2001), is proposed to deal with medical pattern classification tasks. FAM is a supervised model of Adaptive Resonance Theory (ART) (Carpenter and Grossberg, 1987). It has the ability of overcoming the stability-plasticity dilemma (Carpenter and Grossberg, 1987). The system is *stable* enough to preserve information learned from previous data and is *plastic* enough to learn new information from new data. From the viewpoint of classification and decision support, this capability means that the system can continue to learn knowledge safely *in situ* while providing useful predictions. This behavior is analogous to the behavior of an apprentice (Harrison et al., 2001).

The performance of FAM can be improved if its dynamics are adapted with a search algorithm provided by the GA. However, while the GA is capable of identifying the basins of optima, it is usually unable to exploit the basins effectively to reach the global optimum. Martínez-Estudillo et al. (2006) argued that the lack of precision of the GA search for solutions can be overcome by incorporating the GA with a local search algorithm. The integration of the GA with a local fine-tuning algorithm is called a HGA. Hence, the proposed FAM-HGA model inherits the advantages of capturing information directly from data, and its dynamics are further adapted through global and local search algorithms.

The organization of this paper is as follows. The hybrid FAM-HGA model is detailed in Section 2. In section 3, the classification ability of FAM-HGA is first assessed using the Ripley benchmark data set (Ripley, 1994). To further evaluate the effectiveness of FAM-HGA for medical pattern classification and decision support, a case study using anonymous, real medical records of stroke patients is conducted. A systematic study is carried out to compare the performances of FAM-HGA with those without a local search algorithm (i.e., FAM-GA) and FAM. Implications of FAM-GA in supporting medical decisions are drawn at the end of the paper.

## 2 Fuzzy ARTMAP and Hybrid Genetic Algorithm

### 2.1 Fuzzy ARTMAP (FAM)

FAM is a neural network that consists of a pair of Fuzzy ART modules (i.e., Fuzzy ART$_a$ and Fuzzy ART$_b$) that is linked together through a mapping field. It is capable of self-organizing and self-stabilizing information and network architecture by utilizing a feedforward pass (which is usually employed in most neural network models) in addition to a feedback pass for pattern matching. The network grows with time, subject to a vigilance test (Carpenter et al. 1992), with a user-defined threshold. On presentation of an input pattern, the vigilance test is conducted to evaluate the similarity between the prototypes (weights of the network nodes) and the input pattern. If none of the existing prototypes can satisfy the vigilance test, a new node is introduced, and the input pattern is coded as its prototype. Hence, subject to the vigilance criterion, the number of nodes grows with time, and this leads to the formation of an autonomous network architecture for tackling the problem in hand.

### 2.2 The Genetic Algorithm (GA) and Hybrid GA (HGA)

GAs are stochastic search and optimization algorithms that are based on populations of individuals showing a specific behavior similar to natural selection and evolution (Goldberg 1989). In GAs, the solution of a problem is represented by a finite number of chromosomes, and they are defined and initialized to form a population. Each chromosome represents a potential solution, and is assessed by a fitness function to obtain the fitness value in a generation. Based on the fitness values, some of the chromosomes are selected for genetic operations (i.e., crossover and mutation). A new population (offspring) is created to replace the parent chromosomes in the new generation. The GA process repeats until a user-defined criterion is satisfied.

The GA performs global search by climbing many peaks in parallel, and are likely to guide the search to reach at the most promising region. However, it is less efficient in fine-tuning the search space locally. This is because it uses most of the time in climbing different hills, rather than in refining the solution along a single hill that an optimal point locates (Baskar et al., 2001). Hence, the HGA, which combines the GA with a local search technique, inherits the advantages of both global and local search procedures for enhancing the search qualities. In this paper, the idea of this hybrid search is exploited in the HGA. The GA is first used to perform a global search, and the local search technique is then employed to explore a neighborhood of the solution that has been found through the GA search.

### 2.3 Fusion between FAM and HGA

FAM-HGA is an EANN that undergoes adaptation in two modes: supervised ART learning, and both global and local searches. Training starts with the weight learning process of FAM within a predefined number of maximum epochs. In this regard, FAM performs supervised learning based on a set of training data. The architecture as well as the weights of FAM are tuned autonomously. The HGA is then deployed to enhance the learning capability of FAM by searching and adapting its weights.

The weights of Fuzzy ART$_a$ that represent prototypes of the input data are grouped as a Pittsburg chromosome, $z_0$ , for search and adaptation using the HGA. Note that both weights of Fuzzy ART$_b$ and the mapping field of FAM are not involved in the evolution. The intention is to reinforce the weights of Fuzzy ART$_a$ without affecting its link with Fuzzy ART$_b$. The HGA implements a two-phase search procedure. In first phase, a finite number of chromosomes (i.e., $N_p$) are generated prior to evolution to form a population according to a modified function (Baskar et al., 2001)

$$z_i = z_0 + \mathrm{RMF.*rand}(1, N_p) \qquad (1)$$

where $z_i$ is the $i$-th replicated chromosome of $z_0$; RMF$\in[0,1]$ stands for the range multiplication factor; .* represents scalar multiplication operation; and, rand$(1, N_p)$ $\in[0,1]$ is a uniformly distributed random vector.

Each chromosome is converted into the Fuzzy ART$_a$ module and the fitness of the network is evaluated in terms of classification accuracy, by presentation of training data. Once the fitness values of all chromosomes are calculated, the GA applies Roulette-Wheel to select chromosomes. Crossover and mutation are applied to process the selected chromosomes. In this study, the arithmetic crossover (Baskar et al., 2001) that can deal with real-coded chromosomes (i.e., $z_a, z_b$) is applied, i.e.,

$$z' = \lambda z_a + (1-\lambda) z_b \tag{2}$$

$$z'' = (1-\lambda)\lambda z_a + \lambda z_b \tag{3}$$

where $\lambda\in[0,1]$ is a uniformly distributed random variable; $z'$ and $z''$ are offspring.

Dynamic mutation, as described in Baskar et al. (2001), is adopted. For a given chromosome $z$, if element $z_k$ is selected for mutation, the resulting chromosome is $z' = \lfloor z_1 \cdots z_k^{'} \cdots z_n \rfloor$, where $x_k^{'}$ is randomly selected from either

$$z_k^{'} = z_k + \Delta\left(t, z_k^U - z_k\right) \quad \text{or} \quad z_k^{'} = z_k - \Delta\left(t, z_k - z_k^L\right) \tag{4}$$

where $z_k^U / z_k^L$ are the upper/lower bounds of element $z_k$; $\Delta(t,dx)$ of Eq. (4) is further expanded as

$$\Delta(t,dx) = dx\,.\,r\,.\left(1-\frac{t}{T}\right)^d \tag{5}$$

where $r\in[0,1]$; $T$ is the maximum generation of evolution; and $d$ is a parameter that determines the degree of non-uniformity (usually assumed as 2).

After crossover and mutation are applied, a new set of chromosomes (offspring) is obtained. They are individuals that form a new population. The process of fitness evaluation, selection, and genetic operations on a population of chromosomes is repeated until one of the stopping criteria is satisfied to exit from the phase-1 search, i.e., either an achievement of 100% recognition rate of training data by one of the chromosome or a maximum setting of $T$ generations is reached. If the second stopping criterion is fulfilled, the chromosome of the current population that has the highest fitness value is identified as the weights of the Fuzzy ART$_a$ module.

In second phase of search, the best candidate chromosome obtained from phase-1 of FAM-HGA is identified as an initial point, $z_0$. A population of $N_s$ candidate chromosomes is generated based on $z_0$, i.e.,

$$z_i = z_0 + R_t\,.\,\text{rand}(1, N_s) \tag{6}$$

where $R_t$ =RMF for the first iteration of a pre-defined $LT$ maximum number of iteration of phase-2. The fitness of each candidate chromosome in terms of classification accuracy with the training data is calculated. The chromosome with the highest fitness value ($z_b$) replaces $z_0$, i.e., $z_0 = z_b$, and $R_t$ is decreased by

$$R_{t+1} = R_t * (1-\kappa) \tag{7}$$

where $\kappa$ is a range reduction factor, and is typically set to 0.05 (Baskar et al., 2001). The algorithm either proceeds to calculate the fitness value of each chromosome or is terminated if the following condition is satisfied.

$$\left|z_{b,t} - z_{b,t-1}\right| / \left|z_{b,t}\right| \leq \omega \tag{8}$$

where $\omega$ is a very small positive value, e.g. $\omega = 0.01$ (Baskar et al. (2001); $z_{b,t}$ and $z_{b,t-1}$ represent the best chromosome candidates at iteration $t$ and $t$-1, respectively.

# 3 Experiments

In the following experiments, unless otherwise stated, FAM-HGA was trained with a single epoch by setting the FAM parameters to their "default" values: fast learning rate, $\beta = 1$, and ART$_a$ baseline vigilance parameter, $\bar{\rho}_a = 0.0$, while the HGA parameters were set as follows: the number of replicate chromosomes (i.e., $n$=10), RMF=0.30, crossover rate 0.30, mutation rate 0.01, and $T$=$LT$=10. In addition to FAM-HGA, experiments using FAM-GA and FAM were conducted for performance comparison purposes. Since FAM-GA used only the first phase of evolutionary adaptation of FAM-HGA, the parameter of FAM-GA were the same as those of FAM-HGA, exclude *LT*. As for FAM, the same standard FAM training procedure was adopted with the same $\beta$ and $\bar{\rho}_a$ values aforementioned.

## 3.1 A Benchmark Study Using the Ripley Data Set

The Ripley synthetic data set (Ripley, 1994) is a statistical classification problem. The data for each class are generated from a mixture of two Gaussian distributions with identical covariance matrices. The training and test sets consist of 250 and 1000 samples, respectively, with equal distribution of samples belonging to each class. The optimal Bayes error rate is 8% (or 92% accuracy). This synthetic data is used to evaluate the performances of FAM-HGA, FAM-GA, and FAM. In order to ascertain the stability of the results, the Boostrap method (Efron, 1979) is employed to estimate the 95% confidence intervals of classification accuracy. The 95% confidence interval is used to estimate the average classification accuracy as it is commonly used for parameter estimation (Di Stefano, 2004). In addition, a bootstrap hypothesis test was used to compare the performances (average test accuracy rates) and network size (average number of nodes) at the significance level of 0.05. The alternative hypothesis was that the performance/network size of the first model was lower/smaller than the performance/network size of the second. Hence, the performance/network size of the first model was significantly lower/smaller than the performance/network size of the second if the $p$-value was smaller than 0.05.

As shown in Table 1, all three FAM-based networks cannot achieve accuracy rates close to 92%. This is because FAM-based networks do not have statistical properties; thus cannot cope well with this statistical classification task. However, it is worthwhile to note that FAM-GA and FAM-HGA achieve higher accuracy rates significantly, with fewer nodes, than FAM. Besides, FAM-HGA outperforms FAM-GA, statistically (as indicated by the average accuracy rates estimated at 95% confidence intervals in Table 1, and the $p$-values from the bootstrap hypothesis test for accuracy rates in Table 2). This implies the usefulness of incorporating the local search algorithm into the GA in FAM-HGA.

**Table 1.** Results of the Ripley data set (ACC-Accuracy; #N – number of nodes)

| Network | ACC (%) | 95% Confidence Interval | #N | 95% Confidence Interval |
|---|---|---|---|---|
| FAM (multiple epochs) | 83.40 | [82.64 84.09] | 23.4 | [20.7 27.0] |
| FAM-GA | 86.77 | [86.32 87.18] | 9.9 | [9.5 10.4] |
| FAM-HGA | 87.46 | [87.12 87.79] | 9.9 | [9.5 10.4] |

**Table 2.** Results of the Ripley data set (ACC-Accuracy; #N – number of nodes)

| | | $p$-value | |
|---|---|---|---|
| *Model 1* | *Model 2* | Acc. | #N |
| FAM (multiple epochs) | FAM-GA | 0.000 | 1.000 |
| FAM (multiple epochs) | FAM-HGA | 0.000 | 1.000 |
| FAM-GA | FAM-HGA | 0.005 | 0.490 |

### 3.2 Application to Rankin Category Prediction of Stroke Patients

A set of real, anonymous medical records comprising 661 suspected stroke patients is collected. After consultation with medical experts, a total of 18 features, which include medical history, physical examination, laboratory test results, are extracted from the records to form the input pattern. Further details of the data set can be found in Navarro et al. (2008).

In this study, the objective is to predict the Rankin category of patients upon discharge, i.e., either Class 1 (Rankin scale between 0 and 1, i.e., from no symptoms to no significant disability) or Class 2 (Rankin scale between 2 and 6, i.e., from slight disability to dead) (New and Buchbinder, 2005). The data set consists of 141 records of Class 1 and 520 records of Class 2. In each experiment, the data set is split into two: 2/3 of samples of each class into a training set and the remaining into a test set. The performance indicators are: Accuracy (ratio of correctly classified cases to total number of cases); Sensitivity (ratio of correctly classified positive cases to total number of positive cases); Specificity (ratio of correctly classified negative cases to total number of negative cases). Here, positive and negative cases refer to Class 1 and Class 2, respectively.

The experiment was repeated ten times with different training data sequences. The average accuracy, sensitivity, and specificity rates, along with their 95% confidence intervals, were computed using the bootstrap method (Efron, 1979). Table 3 shows the best classification results of FAM, FAM-GA, and FAM-HGA. Table 4 compares the classification results of the abovementioned networks using the same bootstrap hypothesis test as in section 3.1. Some observations based on the results in Tables 3 and 4 can be summarized as follows.

(1) FAM-GA (ACC=74.75%) outperformed FAM, either single-epoch training (ACC=67.60%) or multi-epoch training (ACC=71.81%). FAM-GA recruited fewer nodes as compared with FAM trained with multi-epoch training. In other words, the classification ability of FAM-GA is better than that of FAM (without adaptation and global search), and with a compact network size.

(2) FAM-HGA (ACC=78.19%) outperformed FAM-GA, while the network sizes of HGA and GA versions were statistically similar. The specificity rate of FAM-HGA was higher than those of FAM-GA and FAM. As such, FAM-HGA is useful in diagnosing suspected stroke patients, especially those in Class 2 (Rankin scale between 2 and 6, i.e., from slight disability to dead).

**Table 3.** The best performances of FAM-based models (ACC-Accuracy; Stdev-Standard Deviation; SENS-Sensitivity; SPEC-Specificity; #N-number of nodes)

| Network | ACC | Stdev | SENS | SPEC | #N | Stdev |
|---|---|---|---|---|---|---|
| FAM (single epoch) | 67.60 | 1.23 | 41.70 | 74.60 | 11.8 | 1.4 |
| FAM (multiple epochs) | 71.81 | 1.38 | 37.23 | 81.15 | 25.1 | 2.6 |
| FAM-GA | 74.75 | 1.58 | 16.60 | 90.46 | 11.8 | 1.4 |
| FAM-HGA | 78.19 | 1.24 | 12.77 | 95.86 | 11.8 | 1.5 |

**Table 4.** Performance comparison with $p$-values using the bootstrap hypothesis test

| | | $p$-value | |
|---|---|---|---|
| *Model 1* | *Model 2* | Acc. | #N |
| FAM (single epoch) | FAM (multiple epochs) | 0.000 | 0.000 |
| FAM (single epoch) | FAM-GA | 0.000 | 0.460 |
| FAM (single epoch) | FAM-HGA | 0.000 | 0.459 |
| FAM (multiple epochs) | FAM-GA | 0.000 | 1.000 |
| FAM (multiple epochs) | FAM-HGA | 0.000 | 1.000 |
| FAM-GA | FAM-HGA | 0.000 | 0.478 |

## 4 Summary

In this paper, the applicability of FAM-HGA to medical pattern classification tasks has been examined. A benchmark problem and a real medical diagnosis problem have been studied. The results have positively indicated that FAM-HGA is able to yield better classification performances than those from FAM and FAM-GA. This is owing to the use of the hybrid evolutionary search algorithm in adapting FAM weights, which has resulted in improvements in the classification performances.

In this work, both FAM-GA and FAM-HGA applies the roulette-wheel selection scheme in their search and adaptation process. For future work, the performance of the FAM-based EANN using alternative selection schemes (e.g., tournament selection) can be conducted. In addition, more experiments with different data sets can be conducted to further ascertain the effectiveness of FAM-HGA as a usable and useful decision support tool in the medical domains.

## References

Abbass, H.A.: An evolutionary artificial neural networks approach for breast cancer diagnosis. Artificial Intelligence in Medicine 25, 265–281 (2002)

Baskar, S., Subraraj, P., Rao, M.V.C.: Performance of hybrid real coded genetic algorithms. International Journal of Computational Engineering Science 2, 583–601 (2001)

Carpenter, G.A., Grossberg, S.: A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics and Image Processing 37, 54–115 (1987)

Carpenter, G.A., Grossberg, S., Rosen, D.: Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks 4, 759–771 (1991)

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J., Rosen, D.: Fuzzy ARTMAP: A neural network architecture for incremental learning of analog multidimensional maps. IEEE Trans. Neural Networks 3, 698–713 (1992)

Efron, B.: Bootstrap methods: another look at the Jackknife. The Annals of Statistics 7, 1–26 (1979)

Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley, Reading (1989)

Harrison, R.F., Cross, S.S., Kennedy, R.L., Lim, C.P., Downs, J.: Adaptive resonance theory: A foundation for "apprentice" systems in clinical decision support. In: Dybowski, R., Gant, V. (eds.) Clinical Applications of Artificial Neural Networks, pp. 192–222. Cambridge University Press, Cambridge (2001)

Martínez-Estudillo, A.C., Hervas-Martinez, C., Martinez-Estudillo, F.J., Garcia-Pedrajas, N.: Hybridization of evolutionary algorithms and local search by means of a clustering method. IEEE Transactions on Systems, Man, and Cybernetics B 36, 534–545 (2006)

Navarro, J.C., et al.: Complication of Acute Stroke: A study in Ten Asian Countries. Neurology Asia 13, 33–39 (2008)

New, P.W., Buchbinder, R.: Critical appraisal and review of the Rankin scale and its derivatives. Neuro Epidemiology 26, 4–15 (2005)

Ripley, D.B.: Neural networks and related methods for classification. Journal of the Royal Statistical Society, Series B 56, 409–456 (1994)

Di Stefano, J.: A confidence interval approach to data analysis. Forest Ecology and Management 187, 173–183 (2004)

Yan, H., Jiang, Y., Zheng, J., Peng, C., Li, Q.: A multilayer perceptron-based medical decision support system for heart disease diagnosis. Expert Systems with Applications 30, 272–281 (2006)

# Coevolutionary Method for Gene Selection and Parameter Optimization in Microarray Data Analysis

Yingjie Hu and Nikola Kasabov

The Knowledge Engineering and Discovery Research Institute,
Auckland University of Technology, New Zealand
{raphael.hu,nikola.kasabov}@aut.ac.nz

**Abstract.** This paper presents a coevolutionary algorithm based personalized modeling (cEAP) for gene selection and parameter optimization for microarray data analysis. The classification of different tumor types is a main application in microarray data analysis and of great importance in cancer diagnosis and drug discovery. However, the construction of an effective classifier involves gene selection and parameter optimization, which poses a big challenge to bioinformatics research. We have explored cEAP algorithm on four benchmark microarray datasets for gene selection and parameter optimization. The experimental results have shown that cEAP is an efficient method for co-evolving complex optimization problems in microarray data analysis.

**Keywords:** Coevolutionary, genetic algorithm, optimization and gene selection.

## 1 Introduction

One of the major applications of DNA microarray technology is to apply the classification analysis on patient samples for cancer diagnosis and prognosis. Currently, global modeling is predominately implemented into clinical decision systems for cancer diagnosis and prognosis. However, many of them cannot successfully lead to reasonable satisfactory disease classification results, due to the huge heterogeneity issue of cancer.

Inspired by the concept of genomic personalized medicine [1], we have introduced the personalized modeling for microarray gene expression data analysis [2]. The basic idea behind the personalized modeling for cancer gene expression data analysis is that the diagnosis and prognosis of cancer patients should be based on the individual patient's specific characteristics represented by their gene expression levels. Moreover, personalized modeling systems are able to create a model specifically for an individual patient which can be further used for the development of tailored treatment.

In order to construct the personalized models for cancer diagnosis and prognosis, it is critical to identify which genes are most important for a specific individual patient, and find the best fit parameters for model construction. Gene

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 483–492, 2009.

selection is a fundamental step to achieve the reliable and reproducible classification results for gene expression data analysis. A typical microarray gene expression dataset contains thousands of genes, but most of them are irrelevant to cancer distinction and often lead to inconsistent classification results. Additionally, appropriate parameter setting is a necessary precursor of the personalized model construction, which involves the appropriate number of patient samples having most similar characteristics to the observed one, the classification threshold for diagnosing the sample (e.g. diseased or healthy, different types of tumors), etc. The aim of this study is to develop an efficient method for gene selection and parameter optimization which are able to lead an optimal or near optimal solution to personalized modeling for microarray analysis.

Evolutionary algorithms(EAs) have been applied in a variety of research fields to search for the optimal solutions in large and complex problem space. The most straightforward way to optimize a given objective is to encode the candidate solutions to a set of binary bit strings that are incorporated into an evolutionary algorithm to convolve towards an optimal solution. However, it is found in our previous personalized modeling experiments that the optimal solution is hard to converge through this kind of encoding method, because the candidate solutions usually require different representations rather than one simple representation, i.e. the optimization problems should be represented in different ways: the representation of gene selection is based on binary bit flipping (indicating the gene either selected or not), while the solution to find the most appropriate parameters for individual patient testing should be real-value encoded.

This paper therefore presents a new method (cEAP) using the concept of coevolutionary algorithm [3] for gene selection and parameter optimization. The remainder of this paper is organized as follows: Section 2 briefly reviews gene selection methods, and coevolutionary algorithms. Section 3 describes the new cEAP algorithm for searching the optimal or near optimal personalized model for gene expression data analysis. In section 4, four benchmark microarray datasets are used to verify the performance of our cEAP method, namely, *colon cancer, leukaemia cancer, lung cancer and ovarian cancer* dataset. Section 5 gives the conclusion and future direction.

## 2 Background and Related Work

### 2.1 Gene Selection

Gene selection addresses the problem of discovering a compact set of genes that can highly represent the objective patterns. With efficient gene selection, the reproducibility of gene expression data experiment is more likely to be guaranteed. The problem of a typical gene selection method in bioinformatics can be formalized as follows: for a given training dataset $D = \{x_{ij}, y_i\} \mid x \in \mathbf{X}, y \in \mathbf{Y}, i = 1 \ldots n, j = 1 \ldots m$, pertaining to a bioinformatics task, the objective of gene selection is generally understood to find a subset of genes that are able to assist a computational model to minimize generalization error. The problem is thus formulated as:

$$\mathcal{F}^* = \operatorname*{argmin}_{\mathcal{F} \in \mathfrak{F}} \mathcal{C}(\mathbf{X}, \mathbf{Y}, \mathbb{I}) \tag{1}$$

where $\mathfrak{F}$ is a family of learning functions (e.g. a classifier), $\mathcal{C}$ denotes the generalization error, $\mathbb{I}$ represents an indicator vector in which each bit indicates whether gene $i$ is selected or not, and $\mathcal{F}^*$ is the finally achieved optimal learning function when the generalization error $\mathcal{C}$ reaches minimal.

A number of computational intelligent methods have been implemented into gene selection for identifying informative genes in microarray research. Simple gene selection methods come from statistical methods, such as t-statistics, Fisher's linear discriminate criterion and Principal Component Analysis (PCA). Recently, more sophisticated methods have been developed, e.g. Significance Analysis of Microarrays (SAM) [4] and Artificial neural networks - evolving connectionist system (ECS) [5]. However, most gene selection methods are developed based on the approach of global modeling, which means one global formula is expected to perform well for all data samples. Due to the heterogeneity of cancer, this type of gene selection methods are not always promising to the disease classification distinction. We have thereby employed the personalized modeling method for gene selection to improve the robustness and efficiency.

### 2.2 Coevolutionary Algorithms

Coevolutionary algorithms(CEAs) have attracted significant attentions as an enhancement and extension of conventional evolutionary algorithms for solving complex computational problems. A general claim of coevolutionary algorithm is an evolutionary algorithm in which the individuals from two or more populations are assigned fitness values based on their interactions with the individuals from the other populations. The difference between CEAs and conventional evolutionary algorithms primarily lies in the evaluation process in which an individual can only be evaluated by having its interaction with evolving individuals (interacting partners).

In CEAs, the whole problem space is decomposed into several subcomponents. The fitness function therefore needs to have the collaborators to recombine all individuals from different subcomponents for evaluation. Based on the fitness value, a best combined individual will be selected as a survivor to produce new generations. The process is iterated until the termination criteria are fulfilled. Coevolutionary algorithms have been successfully implemented into a variety of artificial intelligent models for solving optimization problems, such as neural network coevolution [6] and simple GA based coevolution [7]. Since CEAs are able to deal with the problem composed from different subcomponents, they could be a more appropriate approache for optimizing the problem of multiple representations (e.g. the candidate solutions for gene selection and parameter optimization) in our study.

## 3 The Proposed cEAP Algorithm

The proposed coevolutionary algorithm based personalized modeling method (cEAP) is developed for selecting genes and optimizing the parameters of learning functions(e.g., a classifier threshold) simultaneously. The basic hypothesis underlying cEAP algorithm is that using coevolutionary concept, the searching in multiple subcomponents can converge to an optimal or near-optimal solution that comprises informative genes and appropriate parameters.

Evolution strategy (ES) is an optimization technique using natural problem-dependent representations, and is commonly implemented into coevolutionary algorithms to search the real-valued problem space. Genetic algorithms (GAs) have been widely used in optimization problems in which candidate solutions are encoded by a number of populations of abstract representations. Traditionally, solutions are encoded into binary bit strings. GA is a popular choice for coevolutionary algorithm to evolve the populations in binary bit string subcomponents [6,7]. One concern of using conventional GAs comes from the computational complexity when the size of individuals becomes huge. We employe a compact genetic algorithm (cGA) [8] to search for the optimal solution in the binary-encoded problem space (gene selection), owing to its ability to converge towards the optimum significantly faster comparing to conventional GAs. [8].

### 3.1 Structure of the Populations

In the context of personalized modeling for gene expression data analysis, the whole problem space for the given optimization task is decomposed into three subcomponents as follows:

1. subcomponent $\Omega_{(1)}$ for gene selection that is encoded into a binary bit string, in which each bit denotes wether this gene is selected (1) or not (0),
2. subcomponent $\Omega_{(2)}$ for finding the appropriate number of samples $K$ in the localized problem space, which is real-value encoded, and
3. subcomponent $\Omega_{(3)}$ for determining the classification threshold $\theta$ to best fit the testing of individual patient, which is real-value represented.

The decomposed problem space consisting of three subcomponents for gene selection and parameter optimization is shown in Fig.1

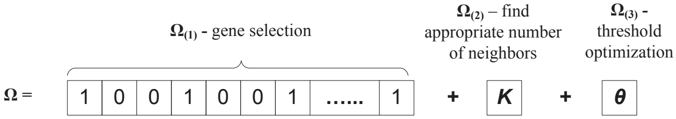

Ω(1) - gene selection
Ω(2) – find appropriate number of neighbors
Ω(3) - threshold optimization
Ω =
1
0
0
1
0
0
1
......
1
+
K
+
θ

**Fig. 1.** The combined individual consisting of 3 subindividuals from subcomponent $\Omega_{(1)}$, $\Omega_{(2)}$ and $\Omega_{(3)}$, respectively

### 3.2 The Principle of cEAP Algorithm

The objective of this study is to build the personalized models for gene expression data analysis, which are able to minimize the prediction accuracy of disease distinction and to create a personalized profile for each individual patient. Given a gene expression data $D = \{X, Y\} \mid X = x_{ij}, Y = y_i, i = 1 \ldots n, j = 1 \ldots m$, the objective is therefore defined to optimize a classifier that involves the selected genes and related parameters:

$$f(s^*) \leq f(s) \tag{2}$$

where $f$ is a classification function of a independent variables set $s$. As $s$ consists of data vector $X$, $Y$ with selected genes and related parameters, Eq.2 can be rewritten as follows:

$$f(X, Y, \zeta_q^*) \leq f(X, Y, \zeta_q), \qquad |\zeta \in \Omega, q = \{1, 2, 3\}. \tag{3}$$

where $\zeta_q$ denotes the candidate solution from $q$ different subcomponents. The final solution is obtained when Eq.3 is fulfilled, i.e. $\zeta_q^*$ is taken as the desired solution to the problem of gene selection and parameter optimization when the classification error is less or equal to the value at any other conditions.

The basic evolving scheme in cEAP for gene selection is based on a compact genetic algorithm (cGA), while an evolutionary strategy (ES) is implemented in the evolution progressed in the subcomponent of parameters optimization. To construct a personalized model for a given dataset $D = \{X, Y\}$ pertaining to the task of cancer diagnosis and prognosis, cEAP algorithm starts with the creation of the population in three subcomponents: gene selection in $\Omega_{(1)}$, number of samples($K$) in $\Omega_{(2)}$ and the disease classification threshold($\theta$) in $\Omega_{(3)}$. The population in gene selection subcomponent is generated based on a probability vector $p$ with $l$ bits ($l \leq n$). Each bit in $p$ is initialized to 0.5, representing the equal probability of this bit(gene) being selected or not. Within the subcomponent $\Omega_{(1)}$, cGA randomly creates two vectors $a$ and $b$, and compare them with the probability vector $p$ in order to generate two bit string individuals $G_a$ and $G_b$. The bit string individual is created based on the comparison result, e.g. if the value of bit $i$ in $a$ is larger than that of bit $i$ in $p$, bit $i$ in $G_a$ is set to 1, otherwise 0. Simultaneously, in the subcomponent $\Omega_{(2)}$, a pair of individuals $K_a$ and $K_b$ are randomly generated by a probability function (e.g. a gaussian distribution function) based on certain domain knowledge. The individuals $\theta_a$ and $\theta_b$ are created in the same way in subcomponent $\Omega_{(3)}$, respectively. Then, subindividuals $G_a$, $K_a$ and $\theta_a$ are recombined into a whole individual $\alpha$ that will be evaluated by a fitness function $\digamma$. Similarly, another candidate individual $\beta$ is created by recombining $G_b$, $K_b$ and $\theta_b$ .

The proposed cEAP algorithm lets individuals $\alpha$ and $\beta$ compete to produce new generations. The evolution in gene selection subcomponent is through updating the property vector $p$ based on the competition result. The updating scheme for $p$ is to check each bit's value of the winner and the loser: if they are same, there is no need to update the $i^{th}$ bit value in vector $p$, otherwise it is

**Algorithm 1.** cEAP algorithm

```
1: initialize the subindividuals in the subcomponent for gene selection:
     generate a probability vector $p$ with $l$ bits, $p_i = 0.5$, where $i \in 1, \cdots, l$,
2: generate two subindividuals from the vector $p$, respectively:
       $(G_a, G_b) = generate(p)$;
3: generate a pair of subindividuals $K_a$, $K_b$ by a probability function $f_p$;
4: generate a pair of subindividuals: $\theta_a$ and $\theta_b$ using a probability function $f'_p$;
5: recombine the above subindividuals from three subcomponents into two individuals:
       $\alpha = G_a + K_a + \theta_a$;
       $\beta = G_b + K_b + \theta_b$;
6: evaluate individuals $\alpha$ and $\beta$ by a fitness function $f$, respectively;
7: compete individual $\alpha$ and $\beta$:
       $winner, loser = compete(\alpha, \beta)$
8: create new populations in three subcomponents:
   (i) use cGA to create the new generation for gene selection subcomponent
       if $G_a(i) \neq G_b(i)$
         if $winner(i) = 1$   then $p_i = p_i + \frac{1}{\mu}$
         else $p_i = p_i - \frac{1}{\mu}$
   (ii) use ES to create the new generation for $K$ and $\theta$ in the other subcomponents:
       Keep the winner of $K$ and $\theta$ to form the offsprings $K_a^{'}$ and $\theta_a^{'}$; the other
       offsprings $K_b^{'}$ and $\theta_b^{'}$ are generated through a mutation performed by probability
       functions $f_p$ and $f'_p$.
9: check wether the termination criteria are reached:
     if yes, then the winner individual represents the final solution $\zeta^*$, including
     the selected genes $G^*$ and optimized parameters $K^*$ and $\theta^*$
     otherwise iterate the process from step 2.
```

updated by $1/\mu$ of increase or decrease, where $\mu$ is the population size. Hence, the new generation created by the updated probability vector $p$ will be more fitted to the fitness function $f$ in principal.

At the same time, evolutionary strategy(ES) algorithm is applied to evolve the new generation in the other subcomponents - $K$ and $\theta$ optimization. A Gaussian function is employed as a probability function to create a new pair of subindividuals for $K$ and $\theta$ according to the result from the competition between $\alpha$ and $\beta$: if the winner's $K$ and $\theta$ are larger than the loser's, then their offsprings should have a higher probability to be larger than their parental pair in the loser. Moreover, the existing domain knowledge can be utilized for parameters initialization, e.g., the most common value for classification threshold $\theta$ is 0.5, and parameter $K$ can be initialized by a ratio - $n/\omega$ ($\omega$ is a weight value and $n$ is the sample size of given data).

Once all the subcomponents have their new generations, cEAP will continue the coevolution and iterate the process until the termination criteria are met. For clarity, the algorithm of cEAP is briefly described by the pseudo code in Algorithm 1.

# 4 Experiment

Four published microarray datasets are used for testing the performance of proposed cEAP method, namely colon cancer data [9], leukaemia data [10], lung cancer data [11] and ovarian cancer data [12].

## 4.1 Experiment Setup

The parameter setting for the experiment is summarized as follows: the initial value of $\theta$ is 0.5 that is the most common threshold used for binary classification problem, and $K = n/\omega$, where $n$ is the sample size of the given data, and $\omega \approx 2.5$. The suggested initial value is based on our previous experimental results for personalized modeling.

Leave-one-out cross validation(LOOCV) is a widely used technique for performance evaluation on small sample size data, which gives an almost unbiased validation result. The sample size in a typical microarray dataset is small, and as a result we take LOOCV classification error estimation as a straightforward approach to evaluate the performance of cEAP method for personalized modeling. For the given data($n$-by-$m$), all samples are divided $n$ times, where in each time all samples except one are used for training and the withheld sample (known as the left out sample) is used for testing. To ensure an unbiased validation, only the training data can be used for gene selection and parameter optimization all through the training process. The outcome (class label) of the leave-one-out testing data keeps unknown to the training process.

The experimental results from cEAP method over four benchmark microarray datasets are applausable in terms of the LOOCV classification accuracy. In order to provide a performance comparison from different methods, we have applied support vector machine (SVM) model and our previously published method using consistency based gene selection algorithm (CAGSC)[13] on these microarray datasets. SVM is generally considered as a reliable and efficient statical classification tool under different conditions. The SVM classifier used in this study is from Neucom (www.theneucom.com), which is an integrated computational intelligent environment for solving complex problems in bioinformatics, data mining, pattern recognition, etc. The parameters of SVM classifier used in this experiment are set as: a polynomial SVM kernal, the number of selected genes is 15. CAGSC method is a consistency based conventional GA approach taht is capable of achieving consistently good classification performance on gene expression data but with high computational cost [14]. It could be a good choice for comparing the efficiency from different optimization methods based on evolutionary computation.

## 4.2 Experiment Result and Discussion

A comparison of classification performance from cEAP, SVM and CAGSC are summarized in Table 1, along with the results reported in the original study of these datasets. It is obvious to elucidate that cEAP method consistently

**Table 1.** The classification accuracy of different methods on all datasets. The classification accuracy of cEAP is presented by overall and class 1/class2 accuracy.

| Dataset | **cEAP**[%] | CAGSC[%] | SVM[%] | original publication[%] |
|---|---|---|---|---|
| **Leukaemia** | 100 (100/100) | 95.84 | 91.17 | 85[10] |
| **Colon** | 87.10 (90.00/81.82) | 83.81 | 83.87 | 87[9] |
| **Lung** | 98.90 (93.55/100) | 95.30 | 95.97 | 90[11] |
| **Ovarian** | 99.60 (100/99.38) | 98.38 | 96.07 | 97[12] |

performs well on these four benchmark datasets. Although SVM obtains reasonably good classification performance, there are some issues that need to be addressed. The parameters used in SVM classifier is not automatically optimized during the learning process, but is prespecified based on previous experimental results or the suggested values in literature. Under such circumstance, it is unfair to a certain extent to compete the other methods with this SVM classifier. Nevertheless, the motivation of using this SVM method is simply to give an overview of a comparison between cEAP and a statical model (SVM) for gene expression data analysis, and to demonstrate the strength of cEAP method in terms of classification error estimation with an unbiased validation approach.

To create a genomic profile for each individual patient, we incorporate cEAP method into our published personalized modeling framework[15]. Due to the limitation of paper length, we only present one example - colon sample#57 for demonstrating the profile of personalized modeling. The proposed cEAP method selects **11** out of 2,000 genes whose expression patterns are most strongly correlated with colon cancer class distinction. Along with these selected genes, two parameters are optimized specifically for sample#57 ($K^* = 24$ and $\theta^* = 0.55$), which leads to a successful disease prediction.

In addition, cEAP creates a scenario of potential genome improvement for the patient(sample#57) which is illustrated in Table 2. The column of actual value in Table 2 represents the actual gene expression level of the given patient sample. Desired average profile is the average gene expression level from the group of healthy people and desired improvement value identifies the change of the gene expression level that the patient should follow in order to recover from disease. For example, the patient(sample#57) should increase the expression level of his/her gene M63391 from 411.6240 to 597.1193. Table 2 also summarizes the importance of each selected genes in terms of the contribution to disease prediction. The Larger the importance value, the more informative the gene.

To validate biology reference of the selected genes by cEAP method, the selected genes from Leukaemia data are compared with the biomarker genes reported in Golub's work[10]. There are 16 genes are most frequently selected by cEAP cross all patient samples to discriminate between acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). These selected genes have shown certain agreement with the reported biomarker genes: 4 of the 16 genes - U22376, M62762,M92287, U32944 are identified informative in both cEAP and

**Table 2.** A scenario of the potential improvement for colon sample#57

| Index of Gene | GenBank Accession Number | Actual value | Desired average profile | Desired Improvement | Weighted importance |
|---|---|---|---|---|---|
| G249 | **M63391** | 411.6240 | 597.1193 | 185.4953 | 0.1241 |
| G377 | **Z50753** | 179.9090 | 233.8870 | 53.9780 | 0.1218 |
| G267 | **M76378** | 397.7460 | 490.9205 | 93.1746 | 0.0970 |
| G419 | **R44418** | 1370.3900 | 249.8221 | -1120.5679 | 0.0942 |
| G1674 | **T67077** | 98.2440 | 56.9415 | -41.3025 | 0.0914 |
| G548 | **T40645** | 717.0060 | 288.2512 | -428.7548 | 0.0903 |
| G1982 | **T89666** | 215.9140 | 43.2651 | -172.6489 | 0.0854 |
| G1582 | **X63629** | 151.1990 | 154.7945 | 3.5955 | 0.0797 |
| G662 | **X68277** | 262.8410 | 428.0565 | 165.2155 | 0.0745 |
| G1870 | **H55916** | 90.0480 | 142.6591 | 52.6111 | 0.0735 |
| G43 | **T57619** | 2997.3980 | 2623.7725 | -373.6255 | 0.0681 |

Golub's method for. The difference of selected informative genes can be accounted by the fact that we use personalized modeling for testing each patient sample while Golub and his colleagues apply a global modeling approach for gene selection. Another interesting finding is that gene U22376 is consistently identified most informative for tumor distinction by both methods. Additionally, this study also concludes that the selected genes for each sample in the same cancer data are not always identical, i.e., the importance of genes for each cancer patient could be varied significantly, even though the genes are known to discriminate between different disease patterns.

## 5 Conclusion and Future Direction

In this study, we have presented a new method(cEAP) using the concept of coevolutionary algorithm for gene selection and parameter optimization in microarray data analysis. Along with our previously proposed personalized modeling framework, we have applied cEAP method on four benchmark cancer gene and protein expression datasets. In terms of classification error estimation, cEAP consistently yields better classification performance. More importantly, cEAP creates the personalized models, including selected genes and optimal disease classification parameters specifically for the observed patient sample, which allows to construct the clinical decision support systems for cancer diagnosis and prognosis.

This work will be further extended to implement other computational intelligence algorithms to enhance the robustness of this algorithm for solving optimization problems, such as ECOS, differential evolutional algorithm, etc. It will also be interesting to cross-check the selected genes from different cancer gene expression data and investigate whether there are some important genes in common for different cancer diagnosis and prognosis.

## References

1. Shastry, B.: Pharmacogenetics and the concept of individualized medicine. Pharmacogenetics 6(1), 16–21 (2006)
2. Kasabov, N.: Global, local and personalized modelling and pattern discovery in bioinformatics: An integrated approach. Pattern Recognition Letters 28(6), 673–685 (2007)
3. Potter, M.A.: The Design and Analysis of a Computational Model of Cooperative Coevolution. PhD thesis, George Mason University (1997)
4. Tibshirani, R.: A simple method for assessing sample sizes in microarray experiments. BMC Bioinformatics 7(106) (2006)
5. Kasabov, N., Middlemiss, M., Lane, T.: A generic connectionist-based method for on-line feature selection and modelling with a case study of gene expression data analysis. In: Conferences in Research and Practice in Information Technology Series: Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics 2003, Adelaide, Australia, vol. 19, pp. 199–202. Australian Computer Society, Inc (2003)
6. Potter, M.A., De Jong, K.A.: Cooperative coevolution: An architecture for evolving coadapted subcomponents. Evolutionary Computation 8(1), 1–29 (2000)
7. Ficici, S., Pllack, J.: A game-theoretic approach to the simple coevolutionary algorithm. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 467–476. Springer, Heidelberg (2000)
8. Harik, G.R., Lobo, F.G., Goldberg, D.E.: The compact genetic algorithm. IEEE Transactions on Evolutionary Computation 3(4), 287–297 (1999)
9. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences of the United States of America 96, 6745–6750 (1999)
10. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mersirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537 (1999)
11. Gordon, G.J., Jensen, R., Hsiao, L.L., Hsiaox, S., Je, B.: Translation of microarray data into clinically relevant cancer diagnostic tests using gege expression ratios in lung cancer and mesothelioma. Cancer Research 62, 4963–4967 (2002)
12. Petricoin, E.F., Ardekani, A.M., Ben A Hitt, P.J.L., Fusaro, V.A., Steinberg, S.M., Mills, G.B., Simone, C., Fishman, D.A., Kohn, E.C., Liotta, L.A.: Use of proteomic patterns in serum to identify ovarian cancer. Lancet 359, 572–577 (2002)
13. Pang, S., Havukkala, I., Hu, Y., Kasabov, N.: Bootstrapping consistency method for optimal gene selection from microarray gene expression data for classification problems. In: Zhang, Y.Q., Rajapakse, J.C. (eds.) Machine Learning for Bioinformatics, pp. 89–111. John Wiley & Sons, Inc., New Jersey (2008)
14. Hu, Y.: Gene Selection Based On Consistency Modelling, Algorithms And Applications - Genetic Algorithm Application. In: Bioinformatics Data Analysis. Vdm Verlag, Saarbrücken (2008)
15. Hu, Y., Song, Q., Kasabov, N.: Personalized modeling based gene selection for microarray data analysis. In: Kóppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008, Part I. LNCS, vol. 5506, pp. 1221–1228. Springer, Heidelberg (2009)

# An Omnibus Permutation Test on Ensembles of Two-Locus Analyses for the Detection of Purely Epistatic Multi-locus Interactions

Waranyu Wongseree[1], Anunchai Assawamakin[2], Theera Piroonratana[1], Saravudh Sinsomros[1], Chanin Limwongse[2], and Nachol Chaiyaratana[1,2]

[1] Department of Electrical Engineering, Faculty of Engineering,
King Mongkut's University of Technology North Bangkok,
1518 Piboolsongkram Road, Bangsue, Bangkok 10800, Thailand
{waranyu.wongseree,theepi,2saravudh,n.chaiyaratana}@gmail.com
[2] Division of Molecular Genetics, Department of Research and Development,
Faculty of Medicine Siriraj Hospital, Mahidol University,
2 Prannok Road, Bangkoknoi, Bangkok 10700, Thailand
anunchai_ice@yahoo.com, siclw@mahidol.ac.th

**Abstract.** Purely epistatic multi-locus interactions cannot generally be detected via single-locus analysis in case-control studies of complex diseases. Recently, many two-locus and multi-locus analysis techniques have been shown to be promising for the epistasis detection. However, exhaustive multi-locus analysis requires prohibitively large computational efforts when problems involve large-scale or genome-wide data. Furthermore, there is no explicit proof that a combination of multiple two-locus analyses can lead to the correct identification of multi-locus interactions. 2LOmb which performs an omnibus permutation test on ensembles of two-locus analyses is proposed. The algorithm consists of four main steps: two-locus analysis, a permutation test, global $p$-value determination and a progressive search for the best ensemble. 2LOmb is benchmarked against a set association approach, a correlation-based feature selection technique and a tuned ReliefF technique. The simulation results from multi-locus interaction problems indicate that 2LOmb has a low false-positive error. Moreover, 2LOmb has the best performance in terms of an ability to identify all causative single nucleotide polymorphisms (SNPs), which signifies a high detection power. 2LOmb is subsequently applied to type 1 and type 2 diabetes mellitus (T1D and T2D) data sets, which are obtained as a part of the UK genome-wide genetic epidemiology study by the Wellcome Trust Case Control Consortium. After primarily screening for SNPs that locate within or near candidate genes and exhibit no marginal single-locus effects, the T1D and T2D data sets are reduced to 2,359 SNPs from 350 genes and 7,065 SNPs from 370 genes, respectively. The 2LOmb search reveals that 28 SNPs in 21 genes are associated with T1D while 11 SNPs in four genes are associated with T2D. The findings provide an alternative explanation for the aetiology of T1D and T2D in a UK population.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 493–502, 2009.

**Keywords:** bioinformatics, epistasis, genetic association study, genetic epidemiology, machine learning.

## 1 Introduction

Complex diseases cannot generally be explained by Mendelian inheritance [1] because they are influenced by gene-gene and gene-environment interactions. Many common diseases such as asthma, cancer, diabetes, hypertension and obesity are widely accepted and acknowledged to be results of complex interactions between multiple genetic factors [2]. Attempts to identify factors that could be the causes of complex diseases have led to many genome-wide association studies [3,4]. Raw results from these attempts produce a large amount of single nucleotide polymorphism (SNP) data from every individual participating in the trials.

For genetic epidemiologists, data sets from genome-wide association studies present many challenges, particularly the correct identification of SNPs that associate with the disease of interest from all available SNPs [5]. This challenge can be treated as a pattern recognition problem which aims to identify an attribute or SNP set that can lead to the correct classification of recruited samples. Heidema et al. [5] and Motsinger et al. [6] have reviewed and identified many machine learning techniques that are suitable to the task. Among many strategies and techniques, the protocol that appears to be most promising for genome-wide association studies involves two main steps: SNP set reduction and classification model construction [7].

The success of the two-step pattern recognition approach relies heavily on the attribute selection step [8]. In case-control studies, epistatic effects play a vital role in establishing the difficulty level of SNP screening problems [9]. Epistasis in the simplest form can be represented by disease models that require genotype inputs from two interacting SNPs [10,11]. Many attempts have been made to produce consistent definitions and categorisation of different types of epistasis models [2,9,12,13,14]. According to Musani et al. [2], a pure epistasis model [15] is difficult because each SNP exhibits no marginal single-locus effect in the model. As a result, it is impossible to detect the pure epistasis by univariate statistical tests.

Many genetic association studies reveal that various complex diseases are results of putative multi-locus interactions [16,17]. With the constraints on a computational capability, exhaustive multi-locus analysis in large-scale or genome-wide association studies would be infeasible [18]. On the other hand, single-locus analysis would be unsuitable for the detection of pure epistasis. One possible approach that provides a trade-off between a computational limitation and an epistasis detection capability is to capture a multi-locus interaction by combining multiple results from two-locus analysis. To achieve this, it is necessary to prove that once a multi-locus interaction model is broken down into a combination of two-locus models, all or some of these models remain detectable through two-locus analysis. Although it is hinted in an early work on two-locus analysis [18] that the proposed approach is plausible, explicit experimentation and testing has never been conducted.

In this article, the feasibility of employing an ensemble of two-locus analyses for the multi-locus interaction determination is demonstrated. Specifically, the significance of the two-locus analysis ensemble is assessed by an omnibus permutation test. The primary function of the proposed method is to detect possible association and assess its significance through the exploration of different ensembles of two-locus analyses. Hence, the proposed method is equally interested in both ensemble selection and testing for significant association.

## 2 Algorithm

The proposed algorithm performs an omnibus permutation test on ensembles of two-locus analyses and is referred to as a 2LOmb technique. The algorithm consists of four steps and can be described as follows.

### 2.1 Two-Locus Analysis

Consider a case-control genetic association study with $n_m$ SNPs, for each pair of SNPs, a $2 \times 9$ contingency table with rows for disease status and columns for genotype configurations is created. A $\chi^2$ test statistic and the corresponding $p$-value can subsequently be computed. With the total of $n_m$ SNPs, there are $\binom{n_m}{2} = n_m!/((n_m - 2)!2!)$ possible SNP pairs. As a result, the $p$-value from each two-locus analysis must be adjusted by a Bonferroni correction. The Bonferroni-corrected $p$-value from each analysis is the lower value between $\binom{n_m}{2}\times$ the uncorrected $p$-value and one.

### 2.2 Permutation Test

The $p$-value $p_0^e$ for the null hypothesis $H_0^e$ that ensemble $e$—an ensemble of two-locus analyses of interest—is not associated with the disease can be evaluated by a permutation test. To achieve this, a scalar statistic is first computed from a function that combines the Bonferroni-corrected $\chi^2$'s $p$-values of individual two-locus tests. A suitable combining function must (a) be non-increasing in each $p$-value, (b) attain its maximum value when any $p$-value equals to zero and (c) have a finite critical value that is less than its maximum for any significant level greater than zero. In this study, a Fisher's combining function $(-2\sum_i \log(p_i))$ is selected. The $p$-value for the ensemble of two-locus analyses is assessed via a permutation simulation. In each permutation replicate, samples are constructed such that the case/control status of each sample is randomly permuted while the total numbers of case and control samples remain unchanged. A $\chi^2$ contingency table with new entries and a Bonferroni-corrected $p$-value for the two-locus analysis within each permutation replicate are then obtained. This, in turn, leads to a new Fisher's test statistic. Let $T_i^e$ denote the value of Fisher's test statistic obtained for the $i$th permutation replicate, $p_0^e$ is the fraction of permutation replicates with a test statistic greater than or equal to the test statistic obtained from the original case-control data ($T_0^e$). In other words,

$$p_0^e = |\{i : 1 \leq i \leq t, T_i^e \geq T_0^e\}|/t, \tag{1}$$

where $t$ is the number of permutation replicates which is set to 10,000 in this study and $|\cdot|$ denotes the size of a set.

### 2.3 Global $p$-Value Determination

There are many candidate ensembles of two-locus analyses that can be explored. Let $H_0 = \bigcap_{1\leq e\leq E} H_0^e$ be the global null hypothesis that none of $E$ explored ensembles of two-locus analyses is associated with the disease, the test of the global null hypothesis leads to the global $p$-value and provides the genetic association explanation. In step 2, the $p$-value $p_0^e$ for a fixed hypothesis $H_0^e$ is a raw or unadjusted $p$-value. To account for the correlation among multiple hypotheses that have been tested during the exploration through many candidate ensembles, the testing result of the global null hypothesis depends on $p_0^{\min} = \min_e p_0^e$. In other words, the global null hypothesis is rejected if the minimum of the raw $p$-values is sufficiently small. The distribution of $p_0^{\min}$ can again be determined by a permutation simulation. However, a nested simulation is unnecessary since the same set of permutation replicates for the $p_0^e$ determination can be reused in the estimation of the empirical distribution of $p_0^{\min}$. The unadjusted $p$-value for the permutation replicate $i$ of each hypothesis $e$ is thus given by

$$p_i^e = |\{j : 0 \leq j \leq t, j \neq i, T_j^e \geq T_i^e\}|/t. \tag{2}$$

Let $p_i^{\min} = \min_e p_i^e$ be the minimum of unadjusted $p$-values over all explored ensembles of two-locus analyses in the $i$th permutation replicate, the $p$-value for the global null hypothesis $H_0$ is defined by

$$p_{\text{global}} = |\{i : 1 \leq i \leq t, p_i^{\min} \leq p_0^{\min}\}|/t. \tag{3}$$

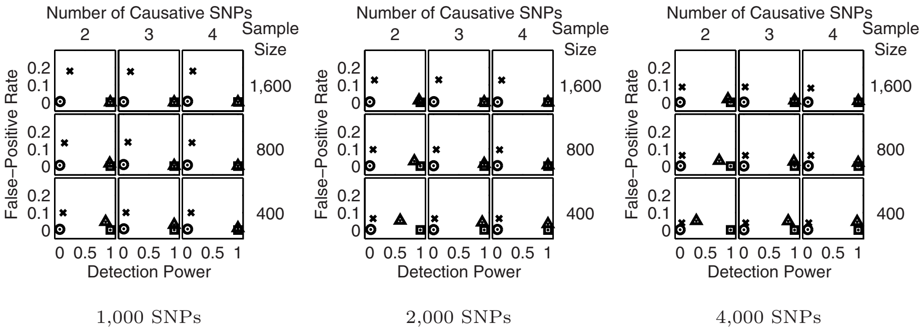
### 2.4 Search for the Best Ensemble of Two-Locus Analyses

A simple progressive search is used to identify the best ensemble of two-locus analyses. The search begins by locating the best two-SNP unit with the smallest Bonferroni-corrected $\chi^2$'s $p$-value from step 1. A permutation test is then performed for this two-locus analysis, yielding both raw and global $p$-values since only one hypothesis has been explored. Next, the search attempts to combine the existing best two-SNP unit with the two-SNP unit that possesses the next smallest Bonferroni-corrected $\chi^2$'s $p$-value from step 1 and does not have a higher permutation $p$-value than the first two-SNP unit. If this new ensemble yields either a higher raw $p$-value or the same raw $p$-value but a higher global $p$-value from a permutation test, the search is terminated and the association is explained by the previously identified two-locus analysis. Otherwise, the best ensemble of two-locus analyses is updated and the process of appending more two-SNP units to the ensemble continues. The progressive search terminates when deterioration in the raw or global $p$-value is detected, or all possible two-locus analyses have been included in the ensemble. It is recalled from step 3 that for the best ensemble containing $E - 1 < \binom{n_m}{2}$ two-locus analyses, its global $p$-value is obtained from the evaluation of $E$ hypotheses.

## 3 Testing with Simulated Data

2LOmb is benchmarked against a set association approach (SAA) [16], a correlation-based feature selection (CFS) technique [8] and a tuned ReliefF (TuRF) technique [19] in a simulation trial. The simulation covers data with causative SNPs, which signify pure epistasis. An efficient algorithm should produce a result with a high number of correctly-identified causative SNPs. This signifies the detection capability. An efficient algorithm should also produces a result with a low number of erroneous SNPs, which are irrelevant to the correct association explanation. This provides an indication for the false-positive error. These two measures on the number of SNPs in the results are used as the performance indicators.

Each simulated data set contains 1,000, 2,000 or 4,000 SNPs in which pure epistasis is governed by two, three or four causative SNPs. The allele frequencies of all causative SNPs are 0.5 while the minor allele frequencies of the remaining SNPs are between 0.05 and 0.5. The data set consists of balanced case-control samples of sizes 400, 800 or 1,600. All SNPs in control samples are in Hardy-Weinberg equilibrium (HWE) [20]. The genotype distribution of causative interacting SNPs follows the pure epistasis model by Culverhouse et al. [15], leading to the heritability of 0.01. Every SNP in each data set exhibits no marginal single-locus effect (Bonferroni-corrected $\chi^2$'s $p$-value $> 0.05$). Seventy-five independent data sets for each simulation setting are generated via a genomeSIM



1,000 SNPs

2,000 SNPs

4,000 SNPs

**Fig. 1.** Performance of SAA, CFS, TuRF and 2LOmb in all simulation scenarios. Detection is declared for SAA and 2LOmb if the $p$-values used as detection indicators in their results are less than 0.05. The results from SAA, CFS, TuRF and 2LOmb are displayed using circle, cross, triangle and square markers, respectively.

package [21]. A paired $t$-test is suitable to assess the significance of results since the same simulated data sets are used during the algorithm benchmarking.

The results from the two-, three- and four-locus interaction problems are shown in Fig. 1. Clearly, 2LOmb significantly outperforms other techniques in terms of the low number of erroneous SNPs, the high number of correctly-identified causative SNPs or both in every interaction problem (a paired $t$-test on 675 benchmark results yields a $p$-value $< 0.05$). The statistical power analysis also reveals that the benchmark trial with 75 independent data sets for each simulation setting is sufficient for an accurate evaluation of the overall algorithm performance (power $> 0.95$ for a Type I error rate of 0.05).

2LOmb also has an advantage in terms of computational time over the other three techniques. The computational time for all four techniques to finish screening the SNPs is provided in Table 1 to demonstrate this strength of 2LOmb. It can be clearly seen that the maximum time required by 2LOmb to screen SNPs in the largest data set is 419 seconds or approximately seven minutes. This is much less than the computational time required by the other techniques for the same data set.

**Table 1.** Computational time required by all four techniques to detect interactions in simulated data sets. Only one computing processor in a Beowulf cluster is occupied during the analysis of one data set. The displayed time is evaluated from the processing of 75 independent data sets for each simulation setting. The computational time from the benchmark trial is the maximum time needed by each method to detect interactions in one data set.

| Number | | Computational Time Required by Each Technique (sec) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| of | | 2LOmb | | SAA | | CFS | | TuRF | |
| Causative | Sample | 1,000 | 4,000 | 1,000 | 4,000 | 1,000 | 4,000 | 1,000 | 4,000 |
| SNPs | Size | SNPs | SNPs | SNPs | SNPs | SNPs | SNPs | SNPs | SNPs |
| 2 | 400 | 15 | 135 | 2,838 | 3,560 | 2,182 | 4,721 | 2,137 | 80,312 |
| | 800 | 21 | 224 | 2,222 | 5,065 | 6,884 | 10,135 | 3,739 | 161,032 |
| | 1,600 | 36 | 400 | 2,997 | 10,105 | 20,054 | 31,034 | 7,134 | 322,084 |
| 3 | 400 | 22 | 140 | 2,788 | 3,539 | 1,302 | 5,546 | 2,319 | 78,892 |
| | 800 | 30 | 229 | 3,239 | 5,059 | 6,888 | 10,328 | 3,521 | 170,936 |
| | 1,600 | 50 | 406 | 5,758 | 10,393 | 19,395 | 30,865 | 5,827 | 322,870 |
| 4 | 400 | 32 | 150 | 3,070 | 5,075 | 3,050 | 6,427 | 2,071 | 73,654 |
| | 800 | 46 | 236 | 3,306 | 6,914 | 8,038 | 12,985 | 3,823 | 157,369 |
| | 1,600 | 70 | 419 | 5,508 | 11,368 | 22,407 | 36,693 | 6,780 | 340,824 |

## 4 Testing with Real Data

2LOmb has been applied to study type 1 and type 2 diabetes mellitus (T1D and T2D) data sets, collected and investigated by the Wellcome Trust Case Control Consortium (WTCCC) [3]. The data sets consist of 2,000 and 1,999 case samples from T1D- and T2D-affected individuals in the UK, respectively and

3,004 shared control samples, which are the results of a merging between 1,500 samples from the UK blood services and 1,504 samples from the 1958 British birth cohort. The original genome-wide data sets contain 500,568 SNPs that are obtained through the Affymetrix GeneChip 500K Mapping Array Set. The SNP sets are primarily reduced by screening for SNPs within or near candidate genes collected by the Human Genome Epidemiology Network (HuGENet) [22]. These candidate genes cover genes from both positive and negative genetic association reports, in which studies are conducted in various ethnic groups and populations. The SNP sets are further reduced by removing SNPs that exhibit strong evidence of genetic association via single-locus analysis. The final T1D and T2D SNP sets contains 2,359 SNPs from 350 genes and 7,065 SNPs from 370 genes, respectively. All SNPs in the reduced data sets exhibit no marginal single-locus effects (Bonferroni-corrected $\chi^2$'s $p$-value $> 0.05$). In the case of T1D, possible genetic association is detected from 28 SNPs in 21 genes (global $p$-value $<$ 0.0001). On the other hand, 11 SNPs in four genes are found to be associated with T2D (global $p$-value $< 0.0001$). Details of these SNPs and the identified genes are given in Tables 2 and 3.

This study produces evidence of positive genetic association. Although there are other independent genome-wide T1D and T2D data sets, the association detection within these data using a similar methodology to the presented method has never been attempted because the methodology employed in the majority of genome-wide association studies is based on single-locus analysis [3]. It is recalled that each SNP explored in the reduced T1D and T2D data sets exhibits no marginal single-locus effect. Hence, the most logical approach to confirm the possibility of replicating association results from the current study is to perform the same detection method on these independent data sets. This is certainly important to gain further understanding of the genetic role in T1D and T2D susceptibility.

## 5 Implementation

2LOmb is implemented in a C programming language. All functions within the program are written by the first author except the $\chi^2$ distribution function, which is taken from the Numerical Recipes in C [23]. The program can be compiled by Microsoft Visual Studio and GNU C compilers. The program has been successfully tested for the execution under Windows and Linux operating systems. The time required by 2LOmb to complete a problem containing $n$ attributes is $T(n) = \binom{n}{2} = n!/((n-2)!2!) = n(n-1)/2$. 2LOmb thus has the order of $n^2$ time complexity ($T(n) \in O(n^2)$). Consequently, 2LOmb can tackle problems in quadratic time. 2LOmb in its present form occupies one processor during the program execution. A parallel version of 2LOmb for genome-wide data is under development. All results included in the study are collected from the execution of computer programs in a Beowulf cluster. The computational platform consists of 12 nodes. Each node is equipped with dual Xeon 2.8 GHz processors and 4GB of main memory. The Rocks Cluster Distribution is installed on all nodes.

**Table 2.** 2LOmb identifies 28 SNPs, which are located in 21 genes, from the reduced T1D data

| Gene | Chromosome and Location | SNP |
|---|---|---|
| *RSBN1* | 1p13.2 | rs6537798 |
| *PTPN22* | 1p13.3–p13.1 | rs3765598 |
| *CCR3* | 3p21.3 | rs13096142 |
| *CCR2* | 3p21.31 | rs1799865 |
| *MOG* | 6p22.1 | rs29228 |
| *NCR3* | 6p21.3 | rs1052248 |
| *AIF1* | 6p21.3 | rs2857697 |
| *BAT2* | 6p21.3 | rs2736172 |
| *APOM* | 6p21.33 | rs805297 |
| *C6orf47* | 6p21.3 | rs2242655 |
| *MSH5* | 6p21.3 | rs2299851, rs707939 |
| *HSPA1L* | 6p21.3 | rs2075800 |
| *TNXB* | 6p21.3 | rs17421624 |
| *AGER* | 6p21.3 | rs1035798 |
| *NOTCH4* | 6p21.3 | rs2071286, rs415929 |
| *C6orf10* | 6p21.3 | rs12524063, rs9268302, rs6907322, rs9268402 |
| *HLA-DQA2* | 6p21.3 | rs9276448 |
| *TAP2* | 6p21.3 | rs1044043 |
| *TAP1* | 6p21.3 | rs4711312, rs12529313 |
| *ITPR3* | 6p21 | rs2229637, rs10947427 |
| *ALDH2* | 12q24.2 | rs7296651 |

**Table 3.** 2LOmb identifies 11 SNPs, which are located in four genes, from the reduced T2D data

| Gene | Chromosome and Location | SNP |
|---|---|---|
| *PGM1* | 1p31 | rs2269241, rs2269239, rs3790857, rs2269238 |
| *LMX1A* | 1q22–q23 | rs2348250, rs6702087 |
| *PARK2* | 6q25.2–q27 | rs1893551, rs6924502 |
| *GYS2* | 12p12.2 | rs6487236, rs1871142, rs10770836 |

## 6 Conclusions

In this article, a method for detecting epistatic multi-locus interactions in case-control data is presented. The study focuses on pure epistasis [2], which cannot be detected via single-locus analysis [15]. To overcome this difficulty, the proposed method performs an omnibus permutation test on ensembles of two-locus analyses and is thus referred to as 2LOmb. The detection performance of 2LOmb is evaluated using both simulated and real data. From the simulation, 2LOmb produces a low false-positive error and a high detection power. Furthermore, 2LOmb outperforms a set association approach (SAA) [16], a correlation-based feature selection (CFS) technique [8] and a tuned ReliefF (TuRF) technique [19] in various interaction scenarios. These scenarios are set up by varying the number of causative SNPs, the number of SNPs in data and the sample size. 2LOmb is subsequently applied to real case-control type 1 and type 2 diabetes mellitus (T1D and T2D) data sets, which are collected from a UK population by the Wellcome Trust Case Control Consortium (WTCCC) [3]. The original genome-wide data sets are first reduced by selecting only SNPs that locate within or near candidate genes reported by the Human Genome Epidemiology Network (HuGENet) [22]. In addition, the selected SNPs must exhibit no marginal single-locus effects. The final T1D data set contains 2,359 SNPs from 350 genes while the final T2D data set contains 7,065 SNPs from 370 genes. 2LOmb identifies 28 SNPs in 21 genes and 11 SNPs in four genes that are associated with T1D and T2D, respectively. This evidence of genetic association leads to an alternative explanation for the aetiology of T1D and T2D in the UK population. It also implies that SNPs from genome-wide data which are usually discarded after single-locus analysis confirms the null hypothesis of no association can still be useful for genetic association studies of complex diseases.

**Acknowledgments.** The authors are grateful to Dr. K. Maneeratana, Dr. P. Tongpadungrod, Dr. G.J. Sheppard, Dr. M. Phadoongsidhi, Dr. V. Varavithya and Dr. S. Tongsima for their valuable suggestions. The authors acknowledge S.M. Dudek and Dr. A. Duncanson for providing an access to the genomeSIM package and the genome-wide case-control data, respectively. WW, AA and TP were supported by the Thailand Research Fund (TRF) through the Royal Golden Jubilee Ph.D. Programme (Grant No. PHD/1.E.KN.49/A.1, PHD/4.I.MU.45/C.1 and PHD/1.E.KN.50/A.1, respectively). CL was supported by the Mahidol Research Grant. NC was supported by the Thailand Research Fund and the National Science and Technology Development Agency through the NSTDA Chair Professor Grant.

## References

1. Risch, N., Merikangas, K.: The Future of Genetic Studies of Complex Human Diseases. Science 273, 1516–1517 (1996)
2. Musani, S.K., Shriner, D., Liu, N., Feng, R., Coffey, C.S., Yi, N., Tiwari, H.K., Allison, D.B.: Detection of Gene × Gene Interactions in Genome-Wide Association Studies of Human Population Data. Hum. Hered. 63, 67–84 (2007)
3. The Wellcome Trust Case Control Consortium: Genome-Wide Association Study of 14,000 Cases of Seven Common Diseases and 3,000 Shared Controls. Nature 447, 661–678 (2007)
4. The GAIN Collaborative Research Group: New Models of Collaboration in Genome-Wide Association Studies: the Genetic Association Information Network. Nat. Genet. 39, 1045–1051 (2007)
5. Heidema, A.G., Boer, J.M.A., Nagelkerke, N., Mariman, E.C.M., van der A, D.L., Feskens, E.J.M.: The Challenge for Genetic Epidemiologists: How to Analyze Large Numbers of SNPs in Relation to Complex Diseases. BMC Genet. 7, 23 (2006)

6. Motsinger, A.A., Ritchie, M.D., Reif, D.M.: Novel Methods for Detecting Epistasis in Pharmacogenomics Studies. Pharmacogenomics 8, 1229–1241 (2007)
7. Moore, J.H., Gilbert, J.C., Tsai, C.T., Chiang, F.T., Holden, T., Barney, N., White, B.C.: A Flexible Computational Framework for Detecting, Characterizing, and Interpreting Statistical Patterns of Epistasis in Genetic Studies of Human Disease Susceptibility. J. Theor. Biol. 241, 252–261 (2006)
8. Hall, M.A., Holmes, G.: Benchmarking Attribute Selection Techniques for Discrete Class Data Mining. IEEE Trans. Knowl. Data Eng. 15, 1437–1447 (2003)
9. Cordell, H.J.: Epistasis: What It Means, What It Doesn't Mean, and Statistical Methods to Detect It in Humans. Hum. Mol. Genet. 11, 2463–2468 (2002)
10. Neuman, R.J., Rice, J.P.: Two-Locus Models of Disease. Genet. Epidemiol. 9, 347–365 (1992)
11. Schork, N.J., Boehnke, M., Terwilliger, J.D., Ott, J.: Two-Trait-Locus Linkage Analysis: a Powerful Strategy for Mapping Complex Genetic Traits. Am. J. Hum. Genet. 53, 1127–1136 (1993)
12. Li, W., Reich, J.: A Complete Enumeration and Classification of Two-Locus Disease Models. Hum. Hered. 50, 334–349 (2000)
13. Marchini, J., Donnelly, P., Cardon, L.R.: Genome-Wide Strategies for Detecting Multiple Loci That Influence Complex Diseases. Nat. Genet. 37, 413–417 (2005)
14. Hallgrímsdóttir, I.B., Yuster, D.S.: A Complete Classification of Epistatic Two-Locus Models. BMC Genet. 9, 17 (2008)
15. Culverhouse, R., Suarez, B.K., Lin, J., Reich, T.: A Perspective on Epistasis: Limits of Models Displaying No Main Effect. Am. J. Hum. Genet. 70, 461–471 (2002)
16. Hoh, J., Wille, A., Ott, J.: Trimming, Weighting, and Grouping SNPs in Human Case-Control Association Studies. Genome Res. 11, 2115–2119 (2001)
17. Heidema, A.G., Feskens, E.J.M., Doevendans, P.A.F.M., Ruven, H.J.T., van Houwelingen, H.C., Mariman, E.C.M., Boer, J.M.A.: Analysis of Multiple SNPs in Genetic Association Studies: Comparison of Three Multi-locus Methods to Prioritize and Select SNPs. Genet. Epidemiol. 31, 910–921 (2007)
18. Gayán, J., González-Pérez, A., Bermudo, F., Sáez, M.E., Royo, J.L., Quintas, A., Galan, J.J., Morón, F.J., Ramirez-Lorca, R., Real, L.M., Ruiz, A.: A Method for Detecting Epistasis in Genome-Wide Studies Using Case-Control Multi-locus Association Analysis. BMC Genomics 9, 360 (2008)
19. Moore, J.H., White, B.C.: Tuning ReliefF for Genome-Wide Genetic Analysis. In: Marchiori, E., Moore, J.H., Rajapakse, J.C. (eds.) EvoBIO 2007. LNCS, vol. 4447, pp. 166–175. Springer, Heidelberg (2007)
20. Hardy, G.H.: Mendelian Proportions in a Mixed Population. Science 28, 49–50 (1908)
21. Dudek, S.M., Motsinger, A.A., Velez, D.R., Williams, S.M., Ritchie, M.D.: Data Simulation Software for Whole-Genome Association and Other Studies in Human Genetics. In: Altman, R.B., Dunker, A.K., Hunter, L., Murray, T., Klein, T.E. (eds.) Proceedings of the Pacific Symposium on Biocomputing 2006, pp. 499–510. World Scientific, Singapore (2006)
22. Yu, W., Gwinn, M., Clyne, M., Yesupriya, A., Khoury, M.J.: A Navigator for Human Genome Epidemiology. Nat. Genet. 40, 124–125 (2008)
23. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C: The Art of Scientific Computing, 2nd edn. Cambridge University Press, Cambridge (1992)

# Protein Fold Prediction Problem Using Ensemble of Classifiers

Abdollah Dehzangi, Somnuk Phon Amnuaisuk, Keng Hoong Ng,
and Ehsan Mohandesi

Artificial Intelligence and Intelligent Computing Center,
Faculty of Information Technology, Multimedia University
Cyberjaya, Selangor, Malaysia
abdollah.dehzangi07@mmu.edu.my, somnuk.amnuaisuk@mmu.edu.my,
khng@mmu.edu.my, e.mohandesi@ece.ut.ac.ir

**Abstract.** Prediction of tertiary structure of protein from its primary structure (amino acid sequence of protein) without relying on sequential similarity is a challenging task for bioinformatics and biological science. The protein fold prediction problem can be expressed as a prediction problem that can be solved by machine learning techniques. In this paper, a new method based on ensemble of five classifiers (Naïve Bayes, Multi Layer Perceptron (MLP), Support Vector Machine (SVM), LogitBoost and AdaBoost.M1) is proposed for the protein fold prediction problem. The dataset used in this experiment is from the standard dataset provided by Ding and Dubchak. Experimental results show that the proposed method enhanced the prediction accuracy up to 64% on an independent test dataset, which is the highest prediction accuracy in compare with other methods proposed by the works have done by literature.

**Keywords:** Protein Fold Prediction Problem, Support Vector Machine, Multi Layer Perceptron, AdaBoost.M1, LogitBoost, Naïve Bayes, Ensemble Method.

## 1 Introduction

Proteins as one of the most significant biological macromolecules have a key role in the life processes. The functions of the protein depend not solely on its amino acid sequence but also heavily effected by its three-dimensional configuration [1]. Prediction of tertiary structure of a protein from its amino acid sequence still remains as an unsolved problem. Recently there was a great interest in applying machine learning approaches to solve this problem. Variety of methods have been implemented and applied to the protein fold prediction problem. In this work, a new ensemble method based on five different classifiers (*Multi Layer Perceptron (MLP), Support Vector Machine (SVM), Naïve Bayes, AdaBoost.M1, LogitBoost*) is proposed to enhance prediction accuracy of the protein fold prediction problem.

Five classifiers are selected based on three criteria: firstly, their prediction accuracies based on previous reports [2], [3], [4] and [5]; secondly, the distribution of the classified proteins depends on the fold and the number of proteins in each fold; and finally, the compatibility of each classifier with other classifiers. In this way the

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 503–511, 2009.

strength of each method is considered and applied to enhance the prediction accuracy, rather than applying each of them separately. Prediction accuracy is measured on the dataset containing proteins belonging to 27 most populated folds from the *Protein Data Bank* (PDB) [6] and *Structural Classification of Proteins* (SCOP) [7] produced by [2]. This dataset was widely applied in later research. The proposed method achieves 64% prediction accuracy on the independent test dataset. This is 3% more than the highest result reported by previous works with the same set of features, found in the literature.

## 2 Literature Review

Recently, machine learning approaches have been widely applied for the protein fold prediction task. But almost all the works were focused on the implementation of unique and sophisticated methods to enhance the prediction accuracy without relying on the other methods advantages. Usually the ensemble methods, introduced to tackle protein folding problem was also based on a single method instead of combination of different methods. Ding and Dubchak classified proteins into 27 fold classes using SVMs and neural networks based on three multi-classification methods (OvO, uOvO, AvA) [2]. They concluded that SVM's performance is better than neural networks.

In contrast to the previous works, in order to deal with the multi-class protein fold classification problem, [8] proposed a *hierarchical learning architecture* (HLA) with automatic feature selection and applied it to the dataset in [2]. At the first level of HLA, neural networks classify the data into the four major Structures of Protein. In the second level, the system has another set of networks, which further classify the data into the 27 folds. The first ensemble method for the protein fold prediction problem was proposed by [3]. They used a 131-dimensional feature vector and an ensemble of four-layer *discretized interpretable multi-layer perceptrons* (DIMLP), where each network learns all protein folds simultaneously.

For the first time an ensemble method based on five different classifiers (MLP, SVM, Naïve Bayes, K-Nearest Neighbor (KNN) and Decision Tree) introduced by [4]. Although later works did not use ensemble methods based on different kind of classifiers but introduced some other unique classifiers that enhanced the accuracy. Recently, [5] applied boosting methods as a kind of ensemble classifier for the protein folding problem. They used AdaBoost.M1 and LogitBoost and achieved to one of the highest prediction accuracies (60%). The highest prediction accuracy achieved by using DIMLP by [3] (61.1%). Despite of all novel and sophisticated methods that have been implemented in previous works, the prediction accuracy was not enhanced adequately.

## 3 Dataset and Features

The training and testing datasets studied here were taken from [2]. The original training dataset is based on the PDB set. This dataset contains 313 proteins with less than 30% sequential similarity of 27 most populated folds. The original test dataset is based on the SCOP database. This dataset contain 385 proteins that have less than 35% sequence similarity. Recently, two proteins (i.e. 2SCMC and 2GPS) in the training dataset and two proteins (2YHX_1 and 2YHX_2) in the testing dataset were removed from this dataset due to the lack of sequence information. Accordingly, new version of dataset contains 311 proteins for training and 383 proteins for testing dataset.

The proteins are alphabetic sequences of amino acids with various lengths. Therefore, to apply machine learning algorithms, we have to transform the amino acid sequences to the numerical feature vectors with equal length. Feature vectors applied in this task are constructed based on physical and chemical properties of amino acids. Six feature vectors were extracted from the sequences: amino acids Composition (C), predicted secondary structure based on Normalized Frequency of $\alpha$ *-helix* (S), *Hydrophobicity* (H), *normalized Van Der Waals volume* (V), *Polarity* (P), and *Polarizability* (Z).

Each feature vector excluding composition contains 21 features. Precisely, composition feature vector represents a vector of the percentage composition of the 20 amino acids in the sequence. More details on how to extract the feature vectors from the sequence of amino acids can be found in [2] and [9]. The length of the amino acid shows its effectiveness for the protein folding task ([3] and [9]); therefore it is included in each combination of features in this work.

## 4 Tools and Methodology

Ensemble methods aim at improving the predictive performance by generating a new method based on construction of linear combination of some proper method, instead of using a single method [10]. Ensemble methods became popular device to improve the predictive performance of base methods [11]. In this work, the ensemble of five methods is used for the protein fold prediction problem to be solved. Each classifier introduced separately in the following sections.

### 4.1 SVM

In this work SVM with the SMO (*Sequential Minimal Optimization*) training algorithm applied. SMO is an algorithm for training a support vector classifier using polynomial kernels [12]. This implementation globally replaces all missing values and transforms nominal attributes into binary ones [13]. To find a decision boundary between two classes a SVM attempts to maximize the margin between the classes, and choose linear separations in a feature space. A function called the kernel $k$ is used to project the data from input space to feature space. The classification of some known point in input space $x_i$ is $y_i$ which is defined to be either -1 or +1. If $x'$ is a point in input space with unknown classification,

$$\mathbf{y}' = Sign\left(\sum_{i=1}^{n} \alpha_i \mathbf{y}_i K(\mathbf{x}_i, \mathbf{x}') + \mathbf{d}\right) \tag{1}$$

Where $y_i$ = {-1, 1} and $y'$ is the predicted class of point $x'$. The function *K* (:) is the kernel, *N* is the number of support vectors, $\alpha_i$ is adjustable weights and $d$ is a bias. Classification time is linear in the number of support vectors.

## 4.2 Naïve Bayes

Naive Bayes classifier is a term in Bayesian statistics dealing with a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model"[14]. The naive Bayes classifier combines Naïve Bayes probability model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the maximum a posteriori or MAP decision rule. The corresponding classifier is the function classify defined as follows:

$$classify(f_1,...,f_n) = \operatorname{argmax}_c \, p(C=c)\prod_{i=1}^{i=n} p(F_i = f_i \mid C=c) \qquad (2)$$

Bayesian networks can efficiently represent complex probability distributions, and have received much attention in recent years [15].

## 4.3 MLP

The most widely used neural classifier today is Multilayer Perceptron (MLP) network which has also been extensively analyzed and for which many learning algorithms have been developed. The MLP belongs to the class of supervised neural networks [16]. A multilayer perceptron is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output.

It is a modification of the standard linear perceptron, which uses three or more layers of neurons (nodes) with nonlinear activation functions. In this paper we apply MLP with one hidden layer and sigmoid activation function [15]. Mathematically this can be written as:

$$y = \varphi(\sum_{i=1}^{n} w_i x_i + b) = \varphi(w^T x + b) \qquad (3)$$

where $w$ denotes the vector of weights, $x$ is the vector of inputs, $b$ is the bias and $\varphi$ is the activation function.

## 4.4 AdaBoost.M1 and LogitBoost

Boosting is a sequential algorithm in which, a base predictor is constructed by applying the given base learner to the training data set with equal weights for each training instance. In the following iterations, the training data with updated weights base on the previously built predictors are provided as the input of the base learner.

The boosting algorithm originally developed for binary classification problems. [17] Extended it to a multi-class case, which they called Adaboost.M1. AdaBoost.M1 is the most straightforward generalization of boosting algorithm [18]. It is adequate when the weak learner is strong enough to achieve high accuracy.

AdaBoostM.1 is fast, simple and easy to program. It requires no prior knowledge about the weak learner, thus can be combined with any method for finding weak hypotheses. AdaBoost.M1 aims to minimize the lost function, especially the exponential loss. [19]

$$\sum_{i=1}^{n} e^{-y_i f(x_i)} \tag{4}$$

LogitBoost performs additive logistic regression to generate a model that maximizes the probability [19]. LogitBoost match a regression model in each iteration data to a weighted training data. For a two class problem, if the weak learner minimizes the squared error, then the probability of the first class is maximized. This rule extended to the multi-class as well. In general, AdaBoost.M1 optimizes the exponential loss where as the LogitBoost minimizes the logistic loss [20].

$$\sum_{i=1}^{n} \log(1 + e^{-y_i f(x_i)}) \tag{5}$$

Data mining toolkit WEKA (Waikato Environment for Knowledge Analysis) version 3.6.0 is used for classification. WEKA is an open source toolkit and it consists of a collection of machine learning algorithms for solving data mining problems [21]. Recently WEKA toolkit widely applied for different classification tasks including protein fold prediction problem.

For AdaBoost.M1, default parameters of WEKA were changed to perform 100 iterations. J48 (WEKA's version of C4.5 [22]) decision tree algorithm was used as its base learner. For LogitBoost default parameters of WEKA were changed to perform 100 iterations and decision stump [22] was used as its base learner. For MLP, SVM and Naïve Bayes default parameters applied.

Each of the methods achieved satisfactory result for the protein fold prediction task. As explained in the literature review SVM and ANN are widely applied for the protein fold prediction problem and showed remarkable results. AdaBoost.M1 achieved 59% and LogitBoost achieved 60.3% prediction accuracy, one of the highest results that reported by the literature [8]. Naïve Bayes achieved 55% accuracy. Considering the simplicity of this methods, made us to use Naïve Bayes in proposed ensemble method.

The results of each classifier show that the MLP and SVM have better result for the folds with larger number of proteins in test and training dataset. On the other hand, AdaBoost.M1 and LogitBoost show better result for the fold with smaller number of proteins in the dataset. Naïve Bayes has almost equal result for all the folds.

From the results, MLP and SVM are used to increase the accuracy for the fold with larger number of proteins. AdaBoost.M1 and LogitBoost are used for the fold with smaller number of proteins. And finally, Naïve Bayes is used in case of judgment between these two groups of classifiers based on the likelihood and posterior probability.

## 5 Results

In this work, ensemble of five classifiers is proposed and evaluated with the dataset that produced by [2]. Proposed method applied with five different voting approaches

(*Maximum Probability, Minimum Probability, Product of Probabilities, Majority Voting and Average of Probability*) (Figure 1) [21]. From the result, Majority Voting [23] shows better prediction accuracy in compare with other voting policies and the previous works (Table1).

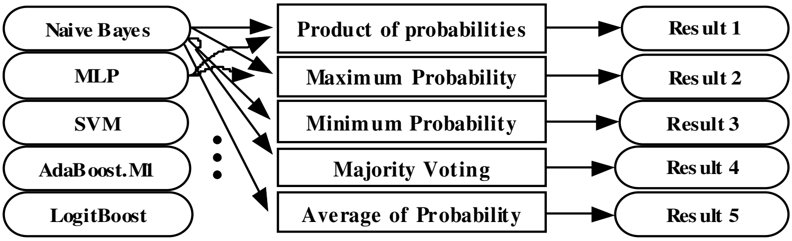


**Fig. 1.** Ensemble of five classifiers with five voting policy

As shown in Table1, results achieved by proposed method by using *Minimum Probability, Product of Probabilities, Majority Voting and Average of Probability* voting policies outperform the results achieved by previous works. Between this four voting policies, the highest prediction accuracy achieved by applying ensemble of all five classifiers with Majority Voting policy for the protein fold prediction problem. The results achieved by this work is not only enhanced the accuracy more than 3%, but it also increased the prediction power of the folds with less proteins in the training and test dataset concurrently.

To assess the effectiveness of each classifier in the achieved prediction accuracy, each method separately omitted of proposed method. The ensemble of remain classifiers with Majority Voting policy evaluated and compared in Table2. As shown in Table2, removing each of the classifiers of proposed method reduced the prediction accuracy (among employed classifiers, removing SVM and MLP respectively show the lowest and the highest prediction accuracy reduction). The ensemble of four methods shows that the results achieved depend on all of the methods that are used in the ensemble method. As mentioned before, each classifier is chosen based on its ability to classify different kinds of protein folds, which depends on the number of proteins in the test and training dataset. Removing of each classifier disturb the balance between classifiers for voting. Base on the results, corporation of all classifiers is necessary to maintain the balance between the results of each classifier for the voting policy.

The results achieved by eleven combination of feature vectors in Table1 and Table2 show the proposed method is not only enhanced the accuracy for combination of all features, but also enhanced the prediction accuracy for other combination of features. the proposed method achieved to 58.48% prediction accuracy that is more than 2% higher than the results (56%) achieved by AvA SVM method proposed by [2] used for combination of three feature vectors.

**Table 1.** The results (in percentage) achieved by ensemble method with five different voting policy in compare various classifiers

| Voting policy | C | CS | CSV | CSZ | CSP | CSH | CSHV | CSHP | CSHPV | CSHPZ | CSHPZV |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Maximum probability | 55.6 | 58.7 | 59.8 | 60.1 | 59.5 | 58.7 | 58.7 | 59.3 | 59.3 | 58.2 | 58.7 |
| Minimum Probability | 57.2 | 57.7 | 57.2 | **62.1** | 60.8 | 59.5 | 58.5 | 60.1 | 59.1 | 59.3 | 57.2 |
| Majority Voting | **58.5** | 61.6 | 61.4 | 62.1 | 60.8 | **62.4** | 59.5 | 59.5 | 62.1 | 61.1 | **64.5** |
| Product of probability | 55.9 | 59.0 | 59.8 | **62.4** | 60.8 | 61.4 | 59.8 | 61.1 | 62.4 | 61.4 | 59.8 |
| Average of Probability | 56.4 | 61.1 | 62.1 | 61.4 | 61.4 | **63.4** | 59.8 | 62.4 | 60.8 | 61.1 | 63.2 |
| AvA SVM [2] | 44.9 | 52.1 | --- | --- | --- | 56.0 | --- | **56.5** | 55.5 | --- | 53.9 |
| OvO SVM [2] | 43.5 | 43.2 | --- | --- | --- | 45.2 | --- | 43.2 | 44.8 | --- | 44.9 |
| uOvO SVM [2] | 49.4 | 48.6 | --- | --- | --- | 51.1 | --- | 49.4 | 50.9 | --- | 49.6 |
| HLA (RBFN) [8] | 44.9 | 52.1 | --- | --- | --- | 56.0 | --- | 56.5 | 55.5 | --- | 53.9 |
| DIMLP [3] | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | **61.1** |
| AdaBoost.M1 [5] | 52.0 | 57.7 | 59.0 | 56.7 | 56.9 | 58.2 | 57.7 | 57.2 | 57.2 | 57.7 | 57.2 |
| LogitBoost [5] | 46.2 | 56.4 | 56.9 | 56.9 | 55.4 | 58.5 | 56.1 | 58.8 | **60.3** | 55.4 | 56.1 |

**Table 2.** The results (in percentage) of ensemble method by omitting each classifier shows in column one

| Excepted Method | C | CS | CSV | CSZ | CSP | CSH | CSHV | CSHP | CSHPV | CSHPZ | CSHPZV |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MLP | 57.1 | 60.3 | **61.6** | 61.6 | 59.0 | 60.6 | 60.6 | 59.5 | 59.5 | 60.6 | 61.6 |
| SVM | 56.1 | 60.6 | 61.4 | 61.4 | 60.1 | 62.4 | 60.8 | 60.8 | 62.9 | 60.1 | **63.7** |
| AdaBoost.M1 | 54.1 | 60.8 | 59.8 | 60.8 | 59.8 | 60.8 | 60.6 | 60.1 | **62.1** | 60.8 | 60.8 |
| LogitBoost | 55.6 | 61.1 | 60.8 | 60.8 | 60.3 | 59.8 | 61.1 | 59.8 | **61.9** | 60.8 | 61.4 |
| Naïve Bayes | 55.9 | 61.9 | 60.8 | 60.8 | 60.6 | 60.1 | 58.0 | 59.3 | 62.7 | 60.8 | **62.9** |

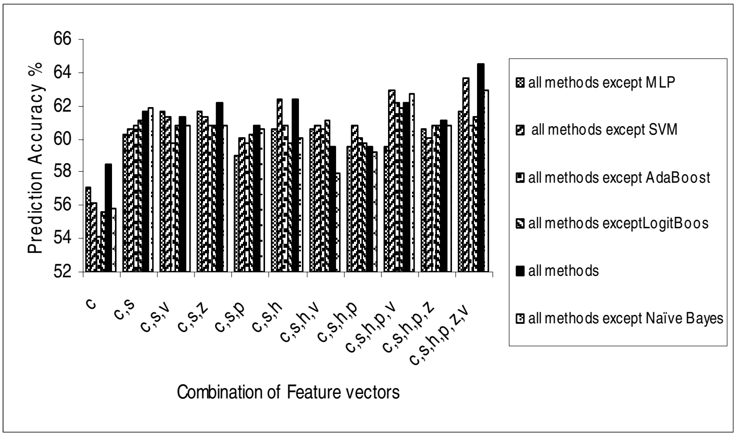


**Fig. 2.** The effectiveness of omitting of each classifier on different combination of feature vectors

The results of omitting each classifier and also combination of all classifiers for eleven combinations of feature vectors are showed and compared in Figure2. The results show the effectiveness of each feature vector on each combination of classifiers. The combination of all features in six cases reached the highest prediction accuracy that shows the compatibility of all feature vectors with other classifiers to achieve the highest prediction accuracy.

## 6 Conclusion

In this paper ensemble of different methods (MLP, SVM, AdaBoost.M1, LogitBoost and Naïve Bayes) was implemented and applied for the protein fold prediction problem. The proposed method achieved 64.5% prediction accuracy which is 3% more than the highest result reported in previous works [3].

Although many powerful methods have been applied to the protein folding prediction problem, but ensemble of different methods has not been paid much attention. Instead of implementing complicated methods based on the strength of one classifier and stochastic voting policies; it is possible to implement ensemble of simpler classifiers. Ensemble of different methods can lead us to inherit merits of each method to enhance the prediction accuracy for different problems especially for the protein fold prediction problem.

## References

1. Stanley Shi, Y.M., Suganthan, P.N.: Multiclass protein fold recognition using multiobjective evolutionary algorithms. In: Computational Intelligence in Bioinformatics and Computational Biology (2004), 0-7803-8728-7
2. Ding, C., Dubchak, I.: Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics 17(4), 349–358 (2001)
3. Bologna, G., Appel, R.D.: A comparison study on protein fold recognition. In: Ninth International Conference on Neural Information Processing, November 2002, vol. 5, pp. 2492–2496 (2002)
4. Bittencourt, V.G., Abreut, M.C.C., de Souto, M.C.P., Canutot, A.M.P.: An empirical comparison of individual machine learning techniques and ensemble approaches in protein structural class prediction. In: International Joint Conference on Neural Networks, 0-7803-9048-2 (2005)
5. Krishnaraj, Y., Reddy, C.K.: Boosting methods for Protein Fold Recognition: An Empirical Comparison. In: IEEE International Conference on Bioinformatics (2008) 978-0-7695-3452-7
6. Hobohm, U., Scharf, M., Schneider, R., Sander, C.: selection of a representative set of structure from the Brookhaven Protein Bank protein. Science 1, 409–417 (1992)
7. Lo Conte, L., Ailey, B., Hubbard, T.J.P., Braner, S.E., Murzin, A.G., Chothia, C.: SCOP a structural classification of proteins database 28(1), 257–259 (2000)
8. Huang, C.D., Lin, C.T., Pal, N.R.: Hierarchical learning architecture with automatic fearture selection for multiclass protein fold classification. IEEE transactions on NanoBioscience 2(4), 221–232 (2003)
9. Duwairi, R., Kassawneh, A.: A Framework for Predicting Proteins 3D Structures. In: Computer Systems and Applications, AICCSA 2008 (2008), 978-1-4244-1968

10. Miller, D.J., Pal, S.: Transductive Methods for the Distributed Ensemble Classification Problem. Neural Computation 19, 856–884 (2007)
11. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms, pp. 1045–9227 (1997)
12. Platt, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: Schoelkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods-Support Vector Learning, 0-262-19416-3. MIT Press, Cambridge (1998)
13. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: a Statistical View of Boosting Annals of Statistics 28(2), 337–407 (2001) (Published version)
14. Friedman, N., Goldszmidt, M.: Learning Bayesian networks with local structure. In: Proc. UAI 1996, pp. 252–262 (1996)
15. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. (2001), 978-0-471-05669-0
16. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. (1998), 978-0-471-05669-0
17. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. Computer Vision and Pattern Recognition (2001), 0-7695-1272-0
18. Schapire, R.E.: The strength of weak learnability. Machine Learning 5, 197–227 (1990)
19. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Annals of Statistics 28(2), 337–407 (2000)
20. Breiman, L.: Bagging Predictors. Machine Learning 24(2), 123–140 (1996)
21. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
22. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
23. MacKay, D.J.C.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press, Cambridge (2003)

# Combination of Multiple Features in Support Vector Machine with Principal Component Analysis in Application for Alzheimer's Disease Diagnosis

Jiann-Der Lee[1,*], Shau-Chiuan Su[1], Chung-Hsien Huang[1], J.J. Wang[2], Wen-Chuin Xu[3] ,You-You Wei[3], and S.T. Lee[4]

[1] Department of Electrical Engineering, Chang Gung University, Taiwan 333
jdlee@mail.cgu.edu.tw
[2] Department of Medical Imaging and Radiological Sciences,
Chang Gung University, Taiwan 333
[3] Department of Neuroscience, Chang Gung Memorial Hospital, Taiwan 333
[4] Center of Medical Augmented Reality, Chang Gung Memorial Hospital, Taiwan 333

**Abstract.** Alzheimer's disease (AD) is a progressively neuro-degenerative disorder characterized by symptoms such as memory loss and cognitive degeneration. In the AD-related research, the volumetric analysis of hippocampus is the most extensive study. However, the segmentation and identification of the hippocampus are highly complicated and time-consuming. Therefore, we designed a MRI-based classification framework to distinguish AD's patients from normal individuals. First, volumetric features and shape features were extracted from MRI data. Afterward, Principle component analysis (PCA) was utilized to decrease the dimensions of feature space. Finally, a SVM classifier was trained for AD classification. With the proposed framework, the classification accuracy is improved from 73.08% or 76.92%, by only using volumetric features or shape features, to 92.31% by using three kinds of volume features and two kinds of shape features.

**Keywords:** Alzheimer's disease, Principle component analysis, SVM.

## 1 Introduction

Alzheimer's disease (AD) is a progressively neuro-degenerative disorder characterized by symptoms such as memory loss and cognitive degeneration, resulting in declining quality of daily life. Up to date, AD affects approximately 26 million people worldwide, and this number may increase fourfold by 2050.

Diagnostic criteria for AD are currently based on clinical and psychometric assessment. The candidate patients are evaluated by neuropsychological tests, which usually focus on their memory and language abilities. For example, the Mini Mental State Examination (MMSE) [1] and Clinical Dementia Rating (CDR) [2] are two most frequently used tests.

[*] Corresponding author.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 512–519, 2009.

Image-based volumetric analysis draws a lot of attention in AD-related research in the past decade, noticeably in regions such as hippocampus [3]. Study shows that the medial temporal lobe structures, especially entorhinal cortex and hippocampus, are highly related with the disease progress [4]. However, the segmentation and identification of hippocampus are usually sensitive to the subjective opinion of the operator and also time-consuming. In the other hand, the enlargement of ventricles is also a significant characteristic of AD [5] due to neuronal loss. Ventricles locate in the center of the brain. They are filled with cerebro-spinal fluid (CSF) which manages the metabolism of the brain and surrounded by gray matter (GM) and white matter (WM). The coverage of GM and WM structures are often affected by dementia diseases. By measuring the extent of ventricular enlargement, the calculated hemispheric atrophy rate can show higher correlation with the disease progression when compared to the medial temporal lobe atrophy rates or on cognitive tests [6], and showed less variation between normal individuals and the subjects with Mild Cognitive Impairment (MCI) and AD [3].

However, selections of region of interest are usually operator-dependent, exhaustive and time-consuming. In this study, we designed an image-based classification framework to distinguish patients with AD from normal individuals using Magnetic Resonance Imaging (MRI). Statistical analysis and experimental results are presented.

In this study, we have designed a MRI-based classification framework to distinguish AD's patients from normal individuals. Section 2 explains the proposed framework comprising system flowchart and selected features. Statistical analysis and experimental results are revealed in Section 3. Finally, the conclusion is included in Section 4.

## 2 Flow Chart and Feature Extraction

Figure 1 illustrates the flowchart of the proposed image-aided AD diagnosis system. First, we register each individual's brain MRI data to a T1-weighted MRI template. This process is also called spatial normalization. Next, the segmentation procedure segment brain tissue such as GM, WM, CSF and cerebral ventricle. Two categories of features, based on global volume and local shape information, are calculated. Mann-Whitney U test is performed to filter out the features with low discriminative power. Principle component analysis (PCA) was used to reduce the dimensions of feature space, the result of which was fed into support vector machine (SVM) as a classifier. The details in each step are explained in the following.

### 2.1 Spatial Normalization of MRI Data

A set of MRI data were registered to a standard spatial coordinate system, in our case, Talairach coordinate system [7]. All 3-D MRI sets were normalized to a T1-weighted MRI template provided by ICBM (International Consortium for Brain Mapping) [8] by using a 12-parameter affine transformation.

### 2.2 Volume Features

A clustering-based segmentation algorithm, provided by Statistical Parametric Mapping (SPM5) [10], was performed to extract probability maps of gray matter (GM),

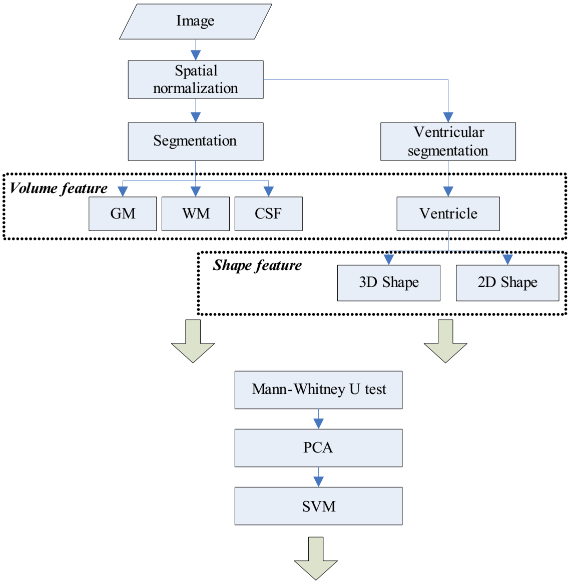


**Fig. 1.** Flowchart of the proposed image-aided AD diagnosis system

white matter (WM) and cerebr-spinal fluid (CSF) from the source MRI data. The value of each pixel in the corresponding probability map indicates the posterior of the pixel belonging to the tissue by giving its gray intensity. The volumes of each tissue type were calculated using the following equations:

$$\text{volume}_{\text{GM}} \approx \sum_{\forall i \in I} (P(C_{gray} \mid f(i)) > 0.5) \tag{1}$$

$$\text{volume}_{\text{WM}} \approx \sum_{\forall i \in I} (P(C_{white} \mid f(i)) > 0.5) \tag{2}$$

$$\text{volume}_{\text{CSF}} \approx \sum_{\forall i \in I} (P(C_{CSF} \mid f(i)) > 0.5) \tag{3}$$

$$\text{volume}_{\text{Whole}} \approx \sum_{\forall i \in I} (P(C_{GM \vee WM} \mid f(i)) > 0.5) \tag{4}$$

where *f(i)* is the gray level of from pixel *i* within data of interest.

In additions, region growing algorithm was used to segment cerebral ventricle due to the sharp contrast between ventricle and surrounding tissue in T1-weighted MRI images. The segmented region constructed a mask image, where 1 stands for the ventricle pixel in mask image and 0 stands for the non-ventricle pixel. At last, using Eq. (5) was employed to measure the cerebral ventricle.

$$\text{volume}_{\text{Ventricle}} \approx \sum_{\forall i \in M} (P(C_{Ventricle} \mid f(i)) = 1) \tag{5}$$

where *f(i)* stands for the gray level of *i* within the mask data M.

### 2.3 Shape Features

In contrast to the volume features, which are extracted from the whole 3-Dimensional volume, the local shape features, such as area, distances between salient points and symmetry, are extracted from a single 2-Dimensional slice [11, 12]. Here, shape features involve 3-D shape feature and 2-D shape features.

In the feature of 3-D shape, the leave-one-out method strategy was used to construct training set and testing set. Then two sets of probability map was build up using Eq. (6) and Eq. (7) for the normal and patients in training set, as shown in Fig. 2 (a) and (b).

$$P_{Normal}(x,y,z) = \frac{1}{M}\sum_{i=1}^{M} I^{i}_{Normal}(x,y,z) \tag{6}$$

$$P_{AD}(x,y,z) = \frac{1}{N}\sum_{i=1}^{N} I^{i}_{AD}(x,y,z) \tag{7}$$

where M is the number of normal controls, N is the number of AD patients and I stands for the grey level of the ventricular mask image.

Following, the discriminate map by subtracting the normal probability map from the AD probability map was obtained, as shown in Fig. 2 (c). Finally, matching coefficient (MC) between each testing input and the discriminate map were calculated by Eq. (8).

$$MC^{i}_{Normal\ \mathrm{or}\ AD} = \sum_{\forall x,y,z} D(x,y,z)T^{i}_{Normal\ \mathrm{or}\ AD}(x,y,z) \tag{8}$$

where D( • ) is the discriminate map and T stands for the testing ventricular mask image.

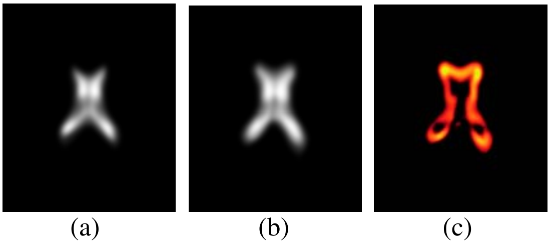


**Fig. 2.** (a) Probability of the normal controls, (b) probability of the AD patients and (c) discriminate map

In 2-D shape features such as distances between salient points are extracted from a single 2-D slice. Therefore, we select the most significant slice from the whole MRI set by the following rules. First, since the CSF is produced from arterial blood by the choroid plexuses of the lateral and fourth ventricles. The axial-view images which comprise choroid plexus are selected as the candidate images. The second rule is that, the ventricle has to form a single connected region. Then, the image which shows maximum area of ventricle and qualifies the second rule from the candidate images was selected.The ways of extract 2-D shape features are explained as follows.

*Perimeter*: Canny edge detection algorithm is used to extract the boundary of the ventricle. The perimeter is equal to the number of boundary pixels.

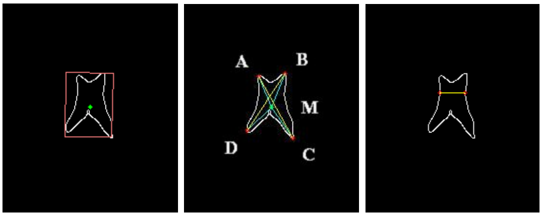
*Compactness*: the square of the perimeter divided by area.

*Elongation*: The ratio of the height and the width of a rotated minimal bounding box which can fit the ventricle, as shown in Fig. 3 (a).

*Rectangularity*: the ratio of the area of an image object and the area of the minimum bounding rectangle.

*Distances*: four corner points on the brain ventricle shape, i.e. the points A, B, C and D shown in Fig. 3 (b). The centroid M of the ventricle was computed. Six different distances was then calculated, i.e. d(A,M), d(B,M), d(C,M), d(D,M), d(A,C) and d(B,D), respectively.

*Minimum thickness*: the minimum distance between path(A,D) and path(B,C), as shown in Fig. 3 (c).



(a) (b) (c)

**Fig. 3.** (a) Minimum bounding box, (b) four corner points, mass point, and (c) minimum thickness

*Mean signature value*: the mean value of the distances from each boundary pixel to the centroid. It starts with the corner point A and follows in the clockwise direction.

# 3 Experimental Results

## 3.1 Material

The image data used in this study were provided by Chang Gung Memorial Hospital, Lin-Kou, Taiwan. The whole dataset consists of 26 individuals (12 males, 14 females) between 55 years and 77 years old (mean age 66.2±6.1 years). Fourteen individuals are normal controls (7 males, 7 females) between 55 years and 77 years of age (mean age 63.1±5.7 years). Twelve individuals were diagnosed as probable AD patients (5 male, 7 female) between 61 years and 75 years old (mean age 69.8±4.3 years). The diagnosis was based on the MMSE complemented by verbal memory, figurative memory and visuospatial tests. The whole-brain MRI scans were obtained from a 3T MR scanner using a T1 MPRage sequence with TR=2000ms and TE=2.63ms. The image matrix is 224×256 and slice thickness is 1mm. One hundred and sixty slices were acquired.

## 3.2 Statistical Analysis and Classification

Mann-Whitney U test was performed on each feature to evaluate its discriminative power. Statistical significance was reach at a threshold when $p < 0.05$. Table 1 showed the statistical results from volume and shape features. In the experiment, the circularity and rectangularity are rejected ($p > 0.05$) in the following steps of classification.

**Table 1.** Statistical analysis of features, where p<0.05 is denoted with a *

| Features | Mean volume in [mm] ± S.D. | | |
|---|---|---|---|
| Volume | Normal | AD | p-value |
| $V_{GM}$ | 847.5±60.1 | 776.6±94.1 | 0.035 |
| $V_{WM}$ | 611.1±53.1 | 534.5±63.9 | 0.003 |
| $V_{CSF}$ | 857.6±137.1 | 969.8±137.8 | 0.021 |
| $V_{Whole}$ | 1458.6±100.2 | 1311.1±145.6 | 0.002 |
| $V_{Ventricle}$ | 40.1±11.7 | 62.6±23.2 | 0.014 |
| Features | Mean ± S.D. | | |
| Shape | Normal | AD | p-value |
| M.C. | 1.51K±0.76K | 3.64K±1.62K | 0.001 |
| Perimeter | 254.4±18.9 | 283.8±36.3 | 0.02 |
| Circularity | 41.9±8.9 | 37.9±6.7 | 0.258 |
| Elongation | 1.2±0.1 | 1.3±0.1 | 0.007 |
| Rectangularity | 0.5±0.1 | 0.6±0.1 | 0.15 |
| d(A,G) | 34.7±3.5 | 39.4±6.0 | 0.031 |
| d(B,G) | 35.9±3.9 | 41.0±4.8 | 0.015 |
| d(C,G) | 37.3±3.1 | 42.9±6.0 | 0.031 |
| d(D,G) | 35.7±3.1 | 40.4±5.6 | 0.02 |
| d(A,C) | 72.1±6.3 | 81.9±11.5 | 0.027 |
| d(B,D) | 70.5±6.7 | 80.9±9.9 | 0.007 |
| Min thickness | 26.9±2.5 | 30.5±4.0 | 0.014 |
| Mean Sig. | 24.5±3.1 | 28.9±4.3 | 0.012 |

PCA was introduced to reduce the dimensions of the feature space in case of redundent or correlated features. The principal components, which contribute 95% to the total variation in the data set, were chosen. To train for a shape-feature-based classification, only the first five principal components were adopted, which convey a large amount of information quantified by 95% energy. In the case of a volume-feature-based classification, all the volume features were adopted. The first six principle components were used when both shape and volume features are used.

For the classification, SVM with RBF kernel was used to train a classifier. SVM is a class of linear classifiers that attempt to achieve maximum margin between the two classes. The classification performance was evaluated by using a leave-one-out trial. Table 2 shows the accuracy (proportion of all subjects correctly classified), sensitivity (proportion of individuals with a true positive result) and specificity (proportion of individuals with a true negative result) when using different features. Obviously, incorporating with shape (M.C and elongation) and volume (CSF, whole and ventricle) features shows excellent classification ability than others. The accuracy, sensitivity and specificity have been improved to 92.31%, 83.33% and 100%, respectively.

**Table 2.** Classification results

| | Volume features | Shape features | Combine |
|---|---|---|---|
| Accuracy | 73.08% | 76.92% | 92.31% |
| Sensitivity | 66.67% | 83.33% | 83.33% |
| Specificity | 78.57% | 71.43% | 100% |

## 4 Conclusions

In this study, a classification framework for image-aided diagnosis for AD was proposed by using easy-extractable volume and shape features. The measurement of global GM, WM and CSF volumes and the local shape analysis on ventricle, especially in the properties of the ventricular area, perimeter and distances, provide atrophy information and show discriminative power that is statistically significant ($p <$ 0.05). The classification accuracy is improved from 73.08% or 76.92%, by only using volumetric features or shape features, to 92.31% by using three kinds of volume features and two kinds of shape features.

## Acknowledgements

This work was supported by Ministry of Economic Affairs, Taiwan under Technology Development Program for Academia (TDPA) with Grant No. 95-EC-17-A-19-S1-035 and Chang Gung Memorial Hospital with Grant No. CMRPD270051.

## References

[1] Hughes, C.P., Berg, L., Danziger, W.L., Coben, L.A., Martin, R.L.: A new clinical scale for the staging of dementia. The British Journal of Psychiatry, 566–572 (1982)

[2] Folstein, M., Folstein, S.E., McHugh, P.R.: Mini-mental state: A practical method for grading the cognitive state of patients for the clinician. Journal of Psychiatric Research 12(3), 189–198 (1975)

[3] Schott, J.M., Price, S.L., Frost, C., Whitwell, J.L., Rossor, M.N., Fox, N.C.: Measuring atrophy in Alzheimer diseases: a serial MRI study over 6 and 12 months. Neurology 65, 119–124 (2005)

[4] Pruessner, J.C., Collins, D.L., Pruessner, M., Evans, A.C.: Age and gender predict volume decline in the anterior and posterior hippocampus. Jornal of Neuroscience 21(1), 194–200 (2001)

[5] Nestor, S., Rupsingh, R., Accomazzi, V., Borrie, M., Smith, M., Wells, J., Bartha, R.: Changes in brain ventricle volume associated with mild cognitive impairment and alzheimer disease in subjects participating in the alzheimer's disease neuroimaging initiative. Alzheimer's and Dementia 3, S114 (2007)

[6] Jack Jr, C.R., Shiung, M.M., Gunter, J.L., O'Brien, P.C., Weigand, S.D., Knopman, D.S.: Comparison of different MRI brain atrophy rate measures with clinical disease progression in AD. Neurology 62, 591–600 (2004)

[7] Talairach, J., Tournoux, P.: Co-Planar Stereotaxic Atlas of a Human Brain: Dimensional Proprotional System: An Approach to Cerebral Imaging. Georg Thieme Verlag, New York (1988)

[8] Mazziotta, J.C., Toga, A., Evans, A., Fox, P., Lancaster, J., Zilles, K.: A probabilistic atlas and reference system for the human brain: International Consortium for Brain Mapping (ICBM). The Royal Society 356(1412), 1293–1322 (2001)

[9] Fritzsche, K.H., von Wangenheim, A., Abdala, D.D., Meinzer, H.P.: A computational method for the estimation of atrophic changes in Alzheimer's disease and mild cognitive impairment. Computerized Medical Imaging and Graphics 32, 294–303 (2008)

[10] UCL Institute of Neurology. Statistical Parametric Mapping (2005), `http://www.fil.ion.ucl.ac.uk/spm/`

[11] Wang, J., Ekin, A., de Haan, G.: Shape Analysis of brain ventricles for improved classification of alzheimer's patients. In: Proceedinds of ICIP, pp. 2252–2255 (2008)

[12] Wang, J., de Haan, G., Unay, D., Soldea, O., Ekin, A.: Voxel-based discriminant map classification on brain ventricles for Alzheimer's disease. Medical Imaging 7259 (2009)

# Hierarchical Core Vector Machines for Network Intrusion Detection

Ye Chen[1], Shaoning Pang[1], Nikola Kasabov[1], Tao Ban[2],
and Youki Kadobayashi[2]

[1] Knowledge Engineering & Discover Research Institute
Auckland University of Technology, Private Bag 92006, Auckland 1020, New Zealand
{zhf0202,spang,nkasabov}@aut.ac.nz
[2] Information Security Research Center, National Institute of Information and Communications Technology, Tokyo, 184-8795 Japan
bantao@nict.go.jp, youki-k@is.aist-nara.ac.jp

**Abstract.** For labelling network intrusions as they state hierarchical multi-label structure, we develop a hierarchical core vector machines (HCVM) algorithm for high-speed network intrusion detection via hierarchical multi-label classification of network data. HCVM models a multi-label hierarchy into a data Hyper-Sphere constructed by numbers of core vector machines (CVM). As the CVMs in an HCVM are separating, encompassing and overlapping with each other, which forms naturally a tree structure representing the multi-label hierarchy encoded. Provided an unlabelled sample, the HCVM seeks a CVM enclosing the sample, and multiply label the sample according to the MEB's position in the hierarchy. The proposed HCVM method has been examined on KDD'99 and the result shows that the proposed HCVM has significant improvement over previously published benchmark works. HCVM improves U2R accuracy from 13.2% to 82.7% and R2L from 8.4% to 45.9%, as compared to the winner of KDD'99. In particular, the efficiency of HCVM is highlighted, as the computational time stays steady while the size of training data exponentially manifolds.

## 1 Introduction

Upon the explosive growth of Internet users, malicious activity such as denial of service attacks, port scans or even attempts to crack into computers by monitoring network traffic are also increasing. These malicious activities on the Internet, also known as network intrusions, trouble the Internet users and cause the Internet security threats. Network intrusion detection systems (NIDS) are developed for safeguarding the users/system from any threats of Internet intrusion. As the preventive measure, existing NIDS mostly employ two popular intrusion detection methods: the white and black list methods filter spam through network ID/address validity verification [1]; and the Intrusion signature recognition methods detect any intrusions by comparing network data with a predetermined attack signature [2]. However, both methods are shown

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 520–529, 2009.

brittle to any small alteration of attack. This requires ordinary NIDS (e.g. a firewall system for personal computer) to be updated in a regular time interval.

As an alternative solution, machine learning methods solve the above difficulty confronted by existing NIDS, because the attack signature/filter are learned dynamically from the streaming network data. In this field, Mukkamala et al. [3] implemented support vector machine (SVM) method on the KDD Cup dataset and obtained an overall 99% classification accuracy. Frank [4] took decision trees as the most suitable classification method for intrusion categorization. Panda and Patra [5] compared the performance of Naive Bayes with the neural network approach, and authenticated the suitability of Naive Bayes for intrusion detection modelling. Despite that, machine learning for network intrusion detection still have underlying issues with aspect to accurate network threats authentication. As seen, with the vulnerability of present-day software and protocols combined with the increasing sophistication of attacks, network-based attacks are on the rise [6], which has made astonishing revenue lost every year. As a fact, the 2005 annual computer crime and security survey [7] reported that the financial losses incurred by the respondent companies due to network attacks/intrusions were US $130 million.

As a status of fact of machine learning for intrusion detection, previous methods perform intrusion detection by grouping any network traffics into several major clusters. This in practice often reduces the detection rate/efficiency because that the detail information of network traffic including attack messages is ignored. For example, spam is simply divided into several major categories, such as junk mail, IM spam, TXT spam etc. It is obviously advantageous to spam detection, if the junk mail is considered further dividable into unsolicited bulk e-mail (UBE) and unsolicited commercial e-mail (UCE). Thus, it is necessary to multiply label the network traffic/intrusions.

In this work, a hierarchy of core vector machines (HCVM) is developed to model the HMC for network intrusion detection, where core vector machine (CVM) is used to enclose a specific type of attacks/intrusions in the smallest hypersphere. Given an unlabeled instance/attack, HCVM seeks a CVM that encloses similar attack type, and labels the sample according to the CVMs position in the hierarchy. HCVM is akin to tree structure, in which the CVMs are separating, encompassing, and overlap with each other. We have authenticated the proposed HCVM classification proficiency and computational efficiency. On comparison with Bernad's method on hierarchical multi-label dataset KDD'99, the proposed HCVM exhibits an outstanding classification accuracy for U2R and R2L attack types. Note that CVM and MEB are two equivalent concept in this paper. In the rest of paper, we use 'CVM' in the context of classification calculation, and 'MEB' for the expression of hypersphere relationship, also we use 'MEB/CVM' for any general description and discussion.

## 2 Related Researches and Motivation

### 2.1 Review of Multi-label Classification

The problem transformation methods which is the most common were multi-label classification method defined in [8] as those methods that transform a multi-label classification problem into either one or multiple single-label's classification or regression [9]. Given dataset $X$ for multi-label classification training, a common problem transformation method is to train $|L|$ binary classifiers $H_l : X \rightarrow \{l, \neg l\}$, one for each individual label $l$ in $L$. Thus, the original data set is transformed into $|L|$ data sets, in which each dataset $D_l$ contains the same instance of the original dataset whose instances are labeled as $l$ or $\neg l$, and a binary classifier is applied for the classification on label $l$. In this way, when an unlabeled instance $x$ is provided for classification, a set of labels are produced simultaneously by $|L|$ classifiers,

$$H(x) = \bigcup_{l \in L} \{l\} : H_l(x) = l. \tag{1}$$

### 2.2 Review of MEB and CVM

The CVM is constructed by the core set of a MEB, which computes the ball(s) of minimum radius enclosing a given set of points. Traditional algorithms for finding exact MEBs developed by Welzl [10] do not scale well with the higher dimensional dataset. Recently, approximation algorithms for finding MEBs have been given by Badoiu [11] and Kumar [12]. Badoiu indicated that an $(1+\epsilon)-approximation$ MEB can be efficiently obtained by using core sets which are subsets of the input dataset for optimizing an approximation MEB. Additionally, Badoiu [11] found that the size of the MEB core set is independent of both dataset dimensionality $d$ and the size of the dataset. Kumar et al. [12] developed methods for computing core sets and approximate the smallest enclosing HyperSpheres in higher dimensions. Tsang [13] proposed core vector machine (CVM) method which constructing a classification model by the corresponding MEB core set in higher dimensions. In this way, the CVM can be implemented in applications requiring solution in relatively higher dimensions.

### 2.3 Motivations of CVM/MEB for HMC

For addressing the HMC problem, we investigate the suitability of CVMs. As mentioned above, a straightforward method is to transfer a HMC problem to a number of single class classification problems using a certain transformation method [9], so that a HMC problem can be managed as a typical multi-class classification. The disadvantage of this method is, the original label hierarchy information (i.e. the partial order of hierarchical multi-labels) of the data is changed or lost completely, which eventually leads to an unsatisfied multi-label classification.

Alternatively, we consider here the HMC problem from the viewpoint of CVM/MEB [13], because for each single class of the HMC problem, wherever it is located in the label hierarchy, it can be approximated by a CVM. One class is represented as one MEB/CVM, meanwhile the CVM may overlap with CVMs from other classes, or encompass CVMs from its child classes. In this way, the HMC problem is addressed by a set of related CVMs, in which the relationship of CVM represents the label hierarchy, so that the original hierarchy information of data get reserved in MEB modeling.

## 3 Methodology

### 3.1 HMC Problem Transformation

Hierarchical multi-label classification (HMC) is an extension of binary classification where an instance can be labeled with multiple classes that are organized in a hierarchy. The definition of HMC task is briefed as follows [14]:
***Given:*** *(1) data set $S = \{x_1, \ldots, x_m\}$, where each $x_i \in \Re^d$, (2) class label set $C = \{c_1, \ldots, c_n\}$, and (3) a class hierarchy $(C, \leq_h)$, where $\leq_h$ is a partial order representing the parent class relationship ($\forall c_1, c_2 \in C : c_1 \leq_h c_2$, if and only if $c_1$ is a superclass of $c_2$).*
***Find:*** *a function $f : x \to C_i$ where $x \in S$ and $C_i \subseteq C$, $C_i$ includes an leaf class $c$ and its parent classes $c'$ such that $c \in C_i \Rightarrow \forall c' \leq_h c : c' \in C_i$.*

### 3.2 Minimum Enclosing Ball

Given a set of points $S = \{x_1, \ldots, x_m\}$, $x_i \in \Re^d$, the minimum enclosing ball of $S$ is the smallest ball that contains all the points in $S$ (denoted by MEB($S$)). Welzl [10] proposed $(1+\epsilon)$-approximation MEB which can be efficiently obtained based on those called Core Set. The core set is a subset of given dataset contains the instances located at the outer area.

Let $B_S(c, r)$ be an exact MEB of the data set $S$ with center $c$ and radius $r$, and $B_Q(\widetilde{c}, \widetilde{r})$ be another exact MEB with center $\widetilde{c}$ and radius $\widetilde{r}$. Note that, different from $B_S$, MEB $B_Q$ is constructed on the Core Set of $S : Q, Q \subset S$. Given an $\epsilon > 0$, a ball $B_Q(\widetilde{c}, (1+\epsilon)\widetilde{r})$ is a $(1+\epsilon)$-approximation of $B_S(c, r)$, if $S \supset B_Q(\widetilde{c}, (1+\epsilon)\widetilde{r})$ and $\widetilde{r} \leq r$.

Formally, subset $Q$ is judged as the core set of $S$, if an expansion by a factor $(1+\epsilon)$ of its MEB contains $S$ (i.e. $S \subset B_Q(\widetilde{c}, (1+\epsilon)\widetilde{r})$. Fig. 1 gives an example of exact MEB, Core set MEB, and Core set MEB expansion, where the dotted circle identifies the exact MEB of the entire dataset $B_S$, and the inside solid line circle gives the exact MEB of Core set $B_Q$ (denoted in square). $B_Q$ does not cover the whole data points, but its $(1+\epsilon)$ expansion (the outside circle) does.

### 3.3 Kernel MEB: Core Vector Machine

By adopting the above MEB algorithm to support vector machine (SVM), Tsang et. al. [13] developed the following kernel MEB method called core vector machine (CVM).

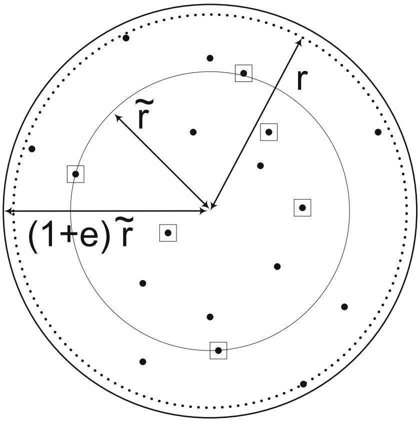


**Fig. 1.** An example of exact MEB (outer circle), Core set MEB (inner circle), and Core set MEB expansion (dotted circle)

Let $\varphi$ be the feature map using kernel $\kappa$, and provided a set of $\varphi$-mapped points $S_\varphi = \{\varphi(x_1), \ldots, \varphi(x_n)\}$, the MEB of $S_\varphi$ is such a smallest ball $B(c^*, r^*)$ that encloses all data points of $S_\varphi$ [13] and has the ball center $c^*$ and radius $r^*$ determined by,

$$(c^*, r^*) = argmin_{c,r} r^2 : \|c - \varphi(x_i)\|^2 \leq r^2 \ \forall i. \tag{2}$$

By the corresponding Wolfe dual,

$$max_{\alpha_i} \sum_{i=1}^{m} \alpha_i k(x_i, x_i) - \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j), \alpha \geq 0, i = 1, \ldots, m, \sum_{i=1}^{m} \alpha_i = 1, \tag{3}$$

where $\alpha = [\alpha_1, \ldots, \alpha_m]$ are the Lagrange multipliers and $K_{m \times m} = [k(x_i, x_j)]$ is the kernel matrix, the MEB center $c$ and radius $r$ can be found from the optimal $\alpha$ as [13],

$$c = \sum_{i=1}^{m} \alpha_i \varphi(x_i), r = -\sqrt{\alpha' diag(K) - \alpha' K \alpha}. \tag{4}$$

In a kernel MEB, a core set instance is called a core vector, and the constructed supervised learning model by core vectors, is called a core vector machine.

### 3.4 Hierarchical Core Vector Machines

For HMC, the above CVM is extended to the approximation of hierarchical core vector machines (HCVM) as follows:

Given dataset $S = (X \times Y)$ where $X \in \Re^d$, and class label set $Y$ is in such a hierarchy as $Y = \{Y_i | y \leq_h \forall y'\}$ whose $y, y' \in Y_i$, and $y' \neq y$. To transform the HMC problem to a HCVM problem, we seek the root class label set $C_1 \subseteq Y$

where $\forall y^l \in C_1 : \neg\exists y' \leq_h y^l : y' \in Y, y' \notin C_1$. For each root class label $y^l \in C_1$, we create an MEB $B$ with the corresponding core set $Q$, mark the MEB and core set with (1) its hierarchical level which is 1, (2) the corresponding class label $y^l$, and (3) its parent class. (we use 0 to represent the parent class of a root class). Thus, for each class $y^l$ in the root class label set $C_1$, we obtain the corresponding MEB $B_1^{y^l,0}$ and the core set $Q_1^{y^l,0}$. Similarly, we address the next level class set $C_2 \subseteq Y$ where $\forall y^l \in C_2 : \exists p^l \leq_h y^l : p^l \in C_{2-1}$, and create the MEB $B_2^{y^l,p^l}$ and the corresponding core set $Q_2^{y^l,p^l}$ for each class label $y^l \in C_2$. This process is continued until the terminal level is reached (i.e. no child classes can be found).

As a result, we have a multi-label hierarchy modelled as a HCVM structured as a set of MEBs ordered with its class label, level in the hierarchy and its parent class,

$$\{Y_i | y \leq_h \forall y'\} \rightharpoonup \{B_1^{y^l,0}, B_2^{y^l,p^l}, ..., B_i^{y^l,p^l}\}. \tag{5}$$

Naturally, the obtained HCVM reserves perfectly the hierarchy information of $Y$. Moreover, it captures in the feature space any invisible relationship between individual classes of the same level, as two MEBs may distribute overlapping despite they reflect two entirely different classes.

For decision making, HCVM assumes that $f(B)$ approximates the ground truth multi-label $f(x)$ better than $f(X, Y)$ because every MEB in HCVM approximates accurately every class of $Y$, meanwhile reserves perfectly the multi-label hierarchy in $Y$. Thus, the HCVM eventually is learned by an aggregation of MEBs as,

$$f(x) = f(\{B_i^{y^l,p^l}\}) = \bigvee_{i=1}^{m} \bigvee_{l=1}^{t_i} f_{B_{i,l}} \tag{6}$$

where $f_{B_{i,l}}$ represents an elementary MEB model on class $y_i^l$. As modelling $f_{B_{i,l}}$, class $y^l$ is addressed as a binary problem with class $y^l$ as the positive and the remaining classes as the negative, and a set of MEBs are computed according to Eq. (2)- Eq. (4) to enclose only the positive instances. Then, $f_{B_{i,l}}$ is formed by assigning the corresponding MEB core sets as support vectors.

In Eq. (6), $\bigvee$ denotes the union between models, in which $\bigvee_i$ represents a natural parent relationship. Given an instance $x$ is enclosed one $i^{th}$-layer MEB, then $x$ due to the parent relationship, it must also be enclosed in the $(i-1)^{th}$, $(i-2)^{th}, ..., 1st, 0$ layer ancestor MEB; and $\bigvee_l$ represents a relationship at the same layer of HCVM. $l$ is often fixed by distances calculation. To compute the distance between $x$ and the center $c_t$ of MEB $B_t$, $x$ is mapped into hyperplane by a kernel function $k(x, x)$. By Eq. (4), the distance between $k(x, x)$ and $c_t$ is calculated as:

$$\|c_t - \varphi(x)\|^2 = \sum_{z_i, z_j \in S_t} \alpha_i \alpha_j k(z_i, z_j) - 2 \sum_{z_i \in S_t} \alpha_i k(z_i, x) + k(x, x). \tag{7}$$

Fig. 2 gives an example of HCVM data approximation over a synthetic dataset with 2 layers hierarchical multi-label $A1 \ \leq_h \ [A2, A3, A4]$, in which $A1$ is the

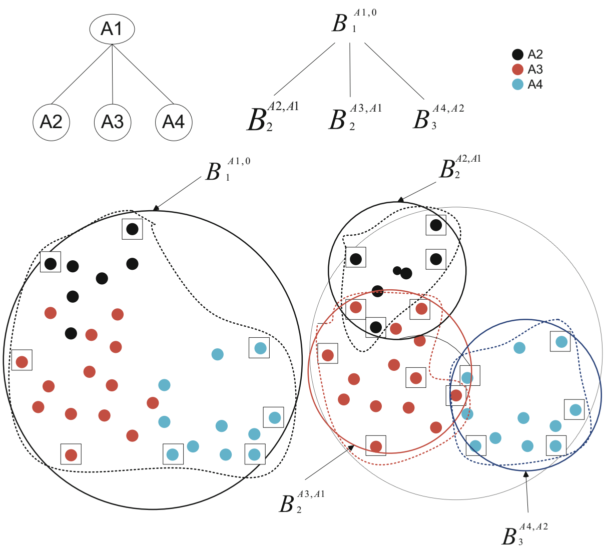


**Fig. 2.** Core set data approximation comparison: boundary on the MEB core set from the parent class A1 versus boundary on the proposed HCVM ($A1 \leq_h [A2, A3, A4]$) core set

parent class and $A2$, $A3$, $A4$ are the child classes of $A1$. To approximate the data distribution of the dataset, an individual MEB can be used to enclose class $A1$ data. As shown in Fig. 2 left, the resulting boundary (identified as the dashed curve) summarizes the knowledge of MEB (represented as the solid line cycle) and the MEB core set (identified as squared data points). Base on the fact that if an instance is in a parent class, then it must belong to one of its child classes, the proposed HCVM method approximates the dataset by combining core sets from $A1$ MEB and MEBs of $A1$'s 3 child classes $A2$, $A3$ and $A4$, and has the boundary for data approximation plotted in Fig. 2 right. It clearly shows that the resulting boundary encloses almost 2 times space than the obtained boundary did. But, no data point exists in the extra space. Thus, the HCVM method gives an optimal minimum enclosing space while discarding the non-essential areas from the MEB for the parent class only.

## 4 Experiments and Discussions

We have implemented the proposed HCVM for the KDD'99 network intrusion detection using MATLAB version(7.6.0), on a 1.86$Hz$ Intel Core 2 machine with 2$GB$ RAM. In our experiment, we use a non-linear Gaussian kernel $k(x, y) = exp(-\|x - y\|^2/\beta)$ with $\beta = \frac{1}{m^2}\sum_{i,j=1}^{m}\left\|x_i - y_j^2\right\|$. We set the $\epsilon$ as $1e-6$ for $(1+\epsilon)$-approximation of MEB, and the Gaussian kernel parameter $\beta$ as 1000, and $t_i$, the parameter representing the number of the nearest MEBs to the input instance, determined by cross validation tests over the training data.

### 4.1 Data Preprocessing

KDD Cup 1999 dataset [15] contains 5 million Internet connection records. Each record encodes 41 connection features including 1 class label, 34 continuous features, and 7 symbolic features. The class label identifies one of 22 network connection types including normal, buffer_overflow, guess_password etc. All connection features are assumed to be conditionally independent. Apart from the normal connection type, the rest of 21 attack types are associated with 4 major categories of attack, they are: (1) DOS, denial of service, e.g. back; (2) R2L, unauthorized access from a remote machine, e.g. guessing password; (3) U2R, unauthorized access to local superuser (root) privileges, e.g. buffer overflow; (4) PROBE, information gathering, e.g. port sweep.

For data preprocessing, we replace the original label for each instance with a set of numerical labels by the following rules: the major attack categories stay at the head followed by its subcategories. For example, 'back', as the first subcategory of DOS attack, is represented as (1,1), and the second subcategory 'land', as (1,2). In this way, all labels can be transformed into a numerical hierarchical structure. Additionally, we normalize every continuous feature into $[0, 1]$, and encode every symbolic feature to binary digit.

### 4.2 Comparison Results

For constructing HCVM, we compute MEBs and core sets for 21 subcategories and the 4 major categories of attacks, respectively. For each category, including subcategories, we use the instances of the core set as the support vectors. As a result, we obtain total 25 support vector sets corresponding to 4 major categories plus 21 subcategories of attacks. Table 1 presents our classification results with a comparison to the results achieved by Bernhard who is the winner of KDD'99 cup.

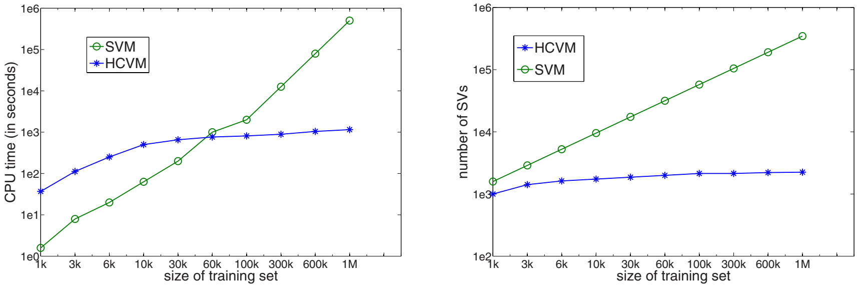
As seen from the table, Bernhard (1999) achieved an extremely high classification accuracy of 99.5% on normal connect type, however U2R and R2L showed poor classification performance, with none of them exceeding 15%, because of their class size being smaller than the other classes. Although the overall classification accuracy of the proposed HCVM is slightly lower than that of the Bernhard's method for normal connection type and Dos type attacks, the proposed HCVM wins Bernhard's method on the classification of 3 most important classes. HCVM particularly increases the classification accuracy of U2R and R2L by 70% and 35%, respectively. This demonstrates the advantage of the proposed HCVM, which encloses all the data points of corresponding class, enabling a more accurate approximation of class information, even for very small dataset.

To evaluate the efficiency of the proposed methods we computed the CPU time, support vectors and test error rate of HCVM and SVM with different size of training dataset. As seen in Fig. 3a, the time cost of HCVM increases more slowly than the SVM. In addition, Fig. 3b shows the number of support vectors found by HCVM almost no change through the increment of training dataset size.

**Table 1.** Compare the classification accuracy with (Bernhard 1999) who is the winner of KDD'99 cup

| Actual VS Predicted | A. Normal | A. DOS | A. U2R | A. R2L | A. Probe |
|---|---|---|---|---|---|
| Bernhard predicted Normal | 60262 | 5299 | 168 | 14527 | 511 |
| HCVM predicted Normal | **920431** | 36818 | 9845 | 2603 | 2084 |
| Bernhard predicted DOS | 78 | 223226 | 0 | 0 | 184 |
| HCVM predicted DOS | 153064 | **3619301** | 10341 | 14732 | 85932 |
| Bernhard predicted U2R | 4 | 0 | 30 | 8 | 0 |
| HCVM predicted U2R | 6 | 3 | **43** | 0 | 0 |
| Bernhard predicted R2L | 6 | 0 | 10 | 1360 | 0 |
| HCVM predicted R2L | 321 | 193 | 44 | **517** | 51 |
| Bernhard predicted Probe | 243 | 1328 | 20 | 294 | 3471 |
| HCVM predicted Probe | 2843 | 2104 | 412 | 506 | **35237** |
| **Bernhard total accuracy** | 99.5% | 97.1% | 13.2% | 8.4% | 83.3% |
| **HCVM total accuracy** | 94.6% | 93.2% | **82.7%** | **45.9%** | **85.7%** |

%



**(a)** CPU time

**(b)** number of support vectors

**Fig. 3.** (a) and (b) plot the CPU time and number of support vectors with different size of training dataset, respectively

## 5 Conclusion

In this paper, we studied a hierarchical multi-label problem with reference to network intrusion detection. By investigating the core set of minimum enclosing ball, which encloses the data points into a smallest HyperSphere, we extended the original MEB algorithm using the Gaussian kernel method, modelling MEBs to be a core vector machine. Further for HMC, we proposed a novel hierarchical CVM methods, where MEBs of CVM separate, encompass and overlap with each other, and construct a hierarchical CVM structure. In our experiments, we implemented the proposed HCVM on benchmark datasets KDD'99. The experimental results show that the proposed HCVM is clearly more favorable than the traditional methods, especially for the U2R and R2L problem.

In general, the proposed HCVM method has the following desirable properties. First, HCVM models a difficult hierarchical multi-label problem into a simple MEB/CVM association analysis. Second, MEB/CVM excludes the existing sparseness of data, which enables HCVM to approximate data more accurately. However, the proposed method inhibits a limitation that the classification performance of the terminal classes might be inferior to that of non-terminal classes since the boundaries for the terminal classes in a multi-label hierarchy are not optimized. As a future work, we will address this limitation by investigating the density of the MEB, and develop new models for HMC.

## References

1. Chirita, P.A., Diederich, J., Nejdl, W.: Mailrank: using ranking for spam detection. In: CIKM 2005: Proceedings of the 14th ACM international conference on Information and knowledge management, pp. 373–380. ACM, New York (2005)
2. Ye, N., Chen, Q.: An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. International 17, 105–112 (2001)
3. Mukkamala, S., Sung, A.H.: Identifying significant features for network forensic analysis using artificial intelligent techniques. Intl. Journal of Digital Evidence 1, 2003 (2003)
4. Frank, J., Mda-c, N.U.: Artificial intelligence and intrusion detection: Current and future directions. In: Proceedings of the 17th National Computer Security Conference (1994)
5. Panda, M., Patra, M.R.: Network intrusion detection using naive bayes. International journal of computer science and network security, 258–263 (2007)
6. Staff, C.: Hackers: companies encounter rise of cyber extortion. Computer Crime Research Center 2006 (2005)
7. CSI, FBI: Proceedings of the 10th annual computer crime and security survey, vol. 10, pp. 1–23 (2005)
8. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining 3(3), 1–12 (2007)
9. Boutell, M.R.: Learning multi-label scene classification. Pattern Recognition 37(9), 1757–1771 (2004)
10. Welzl, E.: Smallest enclosing disks (balls and ellipsoids). In: Maurer, H.A. (ed.) New Results and New Trends in Computer Science. LNCS, vol. 555, pp. 359–370. Springer, Heidelberg (1991)
11. Badoiu, M., Clarkson, K.: Optimal core sets for balls. In: DIMACS Workshop on Computational Geometry (2002)
12. Kumar, P., Mitchell, J.S.B., Yildirim, E.A.: Approximate minimum enclosing balls in high dimensions using core-sets. J. Exp. Algorithmics 8, 1.1 (2003)
13. Tsang, I.W., Kwok, J.T., Cheung, P.-M.: Core vector machines: Fast svm training on very large data sets. Journal of Machine Learning Research 6, 363–392 (2005)
14. Hendrik, B., Leander, S., Jan, S., Amanda, C.: Decision trees for hierarchical multilabel classification: A case study in functional genomics. Journal of Machine Learning Research 4213, 18–29 (2006)
15. KDD 1999 (1999), http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

# String Kernel Based SVM for Internet Security Implementation

Zbynek Michlovský[1], Shaoning Pang[1], Nikola Kasabov[1], Tao Ban[2], and Youki Kadobayashi[2]

[1] Knowledge Engineering & Discover Research Institute
Auckland University of Technology, Private Bag 92006, Auckland 1020, New Zealand
{spang,nkasabov}@aut.ac.nz

[2] Information Security Research Center, National Institute of Information and Communications Technology, Tokyo, 184-8795 Japan
bantao@nict.go.jp, youki-k@is.aist-nara.ac.jp

**Abstract.** For network intrusion and virus detection, ordinary methods detect malicious network traffic and viruses by examining packets, flow logs or content of memory for any signatures of the attack. This implies that if no signature is known/created in advance, attack detection will be problematical. Addressing unknown attacks detection, we develop in this paper a network traffic and spam analyzer using a string kernel based SVM (support vector machine) supervised machine learning. The proposed method is capable of detecting network attack without known/earlier determined attack signatures, as SVM automatically learning attack signatures from traffic data. For application to internet security, we have implemented the proposed method for spam email detection over the *SpamAssasin* and *E. M. Canada* datasets, and network application authentication via real connection data analysis. The obtained above 99% accuracies have demonstrated the usefulness of string kernel SVMs on network security for either detecting 'abnormal' or protecting 'normal' traffic.

## 1 Introduction

Upon computers and Internet being more and more integrated into our common life, higher security requirements have been imposed on our computer and network system. One of many ways that we can take for increasing the security is to use the Intrusion Detection System (IDS). Intrusion detection (system) is a process of monitoring events occurring in a computer system or network and analyzing them on signs of possible incidents. IDS, as described in [10], can be grouped into two categories: statistical anomaly based IDS and signature based IDS. The idea of statistical anomaly IDS is to detect intrusions by comparing traffic with normal traffic model, looking for deviations. Due to the diversity of network traffic, it is difficult to model normal traffic, as we know that a normal email relaying or peer-to-peer queries may also show like with some intrusion traffic characteristics. Moreover, even for abnormal traffic, it does not in fact

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 530–539, 2009.

constitute an intrusion/attack. Hence, anomaly detection often has a high false alarm rate, thus is seldom used in practice. For signature based IDS, network traffic is examined for predetermined attack patterns known as signatures. A signature consists of a string of characters (or bytes). Nowadays many intrusion detection systems also support regular expressions and even behavioral fingerprints [13]. The difficulty of signature based IDS system is that only intrusions whose signatures are known can be detected and it is necessary to constantly update a collection of these signatures to mitigate emerging threats [14].

Another approach for enhancing network security is to authenticate and protect legitimate network traffic. Traditional traffic authentication method based on network protocol (e.g. the port number) is becoming more inaccurate and not appropriate for the identification of P2P and other new types of network traffic. Other methods are mostly based on protocol anomaly detection, which are highly limited because that Internet legitimate traffic does not strictly conform to any specific traffic models. Thus those methods often involve the difficulty of high false positives error in real applications. For example, legitimate *Smtp* traffic could be identified as malicious traffic, if there is misconfiguration in MTA server adding suspicious fields to the header of email message. Thus, it is likely to mis-authenticate valid *Smtp* traffics following just the standard *Smtp* network protocol.

For either attack detection or legitimate network traffic authentication, most signatures/rules are created by security experts who analyze network traffic and host logs after intrusions have occurred, whereas sifting through thousands lines of log files and looking for characteristics that uniquely identify as an intrusion is a vast and error prone undertaking. To overcome this shortcoming and detect unknown attack (i.e. signature is not determined), we researched machine learning on string content recognition techniques. The motivation is to train a classifier to distinguish between malicious and harmless flows or memory dump and utilize the trained classifier to classify real network flow and memory dump.

Support vector machine (SVM) is known as one of the most successful classification algorithms for data mining, in this work we address string, rather than numerical, data analysis, and implemented string kernel SVM for network security in the way of intrusion detection and network application authentication. In spite of the limitation of the SVM on training efficiency, the advantage of string kernel SVMs is in needless of complete knowledge about attack signature/normal application features, as string kernel SVM is able to automatically learn the problem knowledge during the training procedure.

In this work, we develop SVM based string kernel method according to different mathematical similarity expressions of two strings/substrings. For network security, we derive string kernel SVM for automatical attack (i.e. spam emails) signature analysis, conducting spam filtering without early determined spam signature. Moreover, we have used string kernel SVM to authenticate legitimate network applications, learning SVM from connection differences against normal connections.

## 2 SVM and Kernels Theory

Support vector machines (SVM) are groups of supervised learning methods applicable to classification or regression. SVM maps the data points into a high dimensional feature space, where a linear learning machine is used to find a maximal margin separation [7,8]. One of the main statistical properties of the maximal margin solution is that its performance does not depend on the dimensionality of the space where the separation takes place. In this way, it is possible to work in very high dimensional spaces, such as those induced by kernels, without over fitting.

Kernels provide support vector machines with the capability of implicitly mapping non-linearly separable data points into a different higher dimensional space, where they are more separable than the original space. This method is also called *Kernel trick* [7].

Kernel function $K(x, y)$ can be expressed as a dot product in a high dimensional space. If the arguments to the kernel are in a measurable space X, and if the kernel is positive semi-definite for any finite subset $\{x_1, ..., x_n\}$ of X and subset $\{c_1, ..., c_n\}$ of objects

$$\sum_{i,j} K(x_i, x_j)c_i c_j \geq 0, \tag{1}$$

then there must exist a function $\phi(x)$ whose range is in an inner product space of possibly high dimension, such that $K(x, y) = \phi(x)\phi(y)$. The kernel method allows for a linear algorithm to be transformed into a non-linear algorithm. This non-linear algorithm is equivalent to the linear algorithm operating in the range space of $\phi$ . However, because kernels are used, the $\phi$ function is never explicitly computed. The kernel representation of data amounts to a nonlinear projection of data into a high-dimensional space where it is easier to separate into classes [12]. Most popular kernels suitable for SVM are e.g. Polynomial Kernel, Gaussian Radial Basis Kernel, Hyberbolic Tangent Kernel [11]. All of these kernels operate with numerical data. For our purpose is necessary to use string kernels which are described in following section.

## 3 String Kernels Used in SVM

Regular kernels for SVM work merely on numerical data, which is unsuitable for internet security where huge amount of string data is presented. Towards extending SVM for string data processing, we implemented the following string kernels algorithms in our experiments.

### 3.1 Gap-Weighted Subsequence Kernel

The theory of subsequence kernel is described in the book *Kernel Methods for Pattern Analysis* [2]. The main idea behind the gap-weighted subsequence kernel is to compare strings by means of the subsequences they contain - the more subsequences and less gaps they contain the more similar they are. For reducing dimensionality of the feature space we consider non-contiguous substrings that have fixed length $p$. The feature space of gap-weighted subsequence kernel is defined as

$$\phi_u^p(s) = \sum_{i:u=s(i)} \lambda^{l(i)}, u \in \Sigma^p, \tag{2}$$

where $\lambda \in (0,1)$ is decay factor, $i$ is index the occurrence of subsequence $u = s(i)$ in string $s$ and $l(i)$ is length of the string in $s$. We weight the occurrence of $u$ with the exponentially decaying factor $\lambda^{l(i)}$. The associated kernel is defined as

$$\kappa(s,t) = \langle \phi^p(s), \phi^p(t) \rangle = \sum_{u \in \Sigma^p} \phi_u^p(s)\phi_u^p(t). \tag{3}$$

In Eq. (3), it is required to conduct an intermediate dynamic programming table $DP_p$ whose entries are:

$$DP_p(k,l) = \sum_{i=1}^{k}\sum_{j=1}^{l} \lambda^{k-i+l-j} \kappa_{p-1}^S(s(1:i), t(1:j)). \tag{4}$$

Then, the computational complexity is evaluated as

$$\kappa_p^S(sa,tb) = \begin{cases} \lambda^2 DP_p(|s|, |t|) & \text{if } a = b; \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

which follows that for a single value of $p$, the complexity of computing $\kappa_p^S$ is $O(|s|\,|t|)$. Thus, the overall computational complexity of $\kappa_p(s,t)$ is $O(p\,|s|\,|t|)$.

### 3.2 Levenshtein Distance

Levenshtein (or edit) distance [4] counts differences between two strings. The distance is the number of substitutions, deletions or insertions required to transform string $s$ with length $n$ to string $t$ with length $m$. The formal definition of Levenshtein distance [17] is given as follows: Given a string $s$, let $s(i)$ stand for its $i^{\text{th}}$ character. For two characters $a$ and $b$, define

$$r(a,b) = 0 \ if a = b. \ Let \ r(a,b) = 1 \tag{6}$$

Assuming two strings $s$ and $t$ with the length of $n$ and $m$, respectively, then a $(n+1)(m+1)$ array $d$ furnishes the required values of the Levenshtein distance $L(s,t)$.

The calculation of $d$ is a recursive procedure. First set $d(i,0) = i$, $i = 0, 1, ..., n$ and $d(0,j) = j$, $j = 0, 1, ..., m$; Then, for other pairs $i$, $j$, we have

$$d(i,j) = min(d(i-1,j)+1,\ d(i,j-1)+1,\ d(i-1,j-1)+r(s(i),\ t(j))). \tag{7}$$

In our implementation, we use $D = e^{-\lambda \cdot d(i,j)}$ for getting better results. Analog to the above substring kernel, the computational complexity of Levenshtein Distance is $O(|s|\,|t|)$. In the case that $s$ and $t$ have the same length, the complexity is $O(n^2)$.

### 3.3 Bag of Words Kernel

The Bag of words kernel is represented as an unordered collection of words, disregarding grammar and word order. Words are any sequences of letters from the basic alphabet separated by punctuation or spaces. We represent a bag as a vector in a space in which each dimension is associated with one term from the dictionary

$$\phi : d \mapsto \phi(d) = (tf(t_1, d), tf(t_2, d), ....tf(t_N, d)) \in \mathbb{R}^N, \tag{8}$$

where $tf(t_i, d)$ is the frequency of the term $t_i$ in the document $d$. Hence, a document is mapped into a space of dimensionality $N$ being the size of the dictionary, typically a very large number [2].

### 3.4 N-Gram Kernel

N-grams transform documents into high dimensional feature vectors where each feature corresponds to a contiguous substring [5]. The feature space associated with n-gram kernel is defined as

$$\kappa(s, t) = \langle \phi^n(s), \phi^n(t) \rangle = \sum_{u \in \Sigma^n} \phi^n_u(s) \phi^n_u(t) \tag{9}$$

where

$$\phi^n_u(s) = \left| \{ (v_1, v_2) : s = v_1 u v_2 \} \right|, u \in \Sigma^n.$$

We have used for computing n-gram kernel naive approach therefore the time complexity is $O(n \, |s| \, |t|)$.

## 4 Experiments and Discussions

For internet security, one useful approach is to actively detect and filter spam/attack by building Bayesian filters. In addition, another practical approach is to protect legitimate network communication by authenticating every type of legitimate network application. In this section, we implemented string kernel SVMs on the spamAssasin public mail corpus for email spam detection, and experimented the authentication of 12 categories standard network application.

For multi-class classification, we used support vector machine software - *libSVM* [1]. In our experiments, we used precomputed kernel matrices from prepared training and testing datasets. All values in precomputed kernel matrices have been scaled to interval [-1,1]. Optimal parameters of string kernel functions have been determined by cross validation tests on training datasets. Kernel matrices has been applied as input to libSVM [1].

### 4.1 Spam Detection

For spam detection experiment, we used 5500 ham messages from *SpamAssasin public mail corpus* [18] and 24038 spam messages from *E. M. Canada* [19]. The ham emails dataset consists of two categories: *EasyHam emails*, 5000 non-spam messages without any spam signatures and *Hard Ham emails*, 500 non-spam messages similar in many aspect to spam messages - using unusual HTML markup, colored text, spam-sounding phrases, etc. Each email message has a header, a body and some potentially attachments. Note that for the convenience of comparison, we employed the exact same data setup as in [20], training dataset 23630 (19230 spam vs. 4400 ham) messages and testing dataset 5918 (4808 spam vs. 1110 ham) messages.

Analog to [20], we intended to determine which part of email message have critical influence on the classification results. To this end, we prepared four subsets: *Subject*, *Body*, *Header* and *All* subsets. The *Subject* subset uses only the subject field of the email message, and all *Subject* data are normalized to the length of 100 characters; The *body* subset is the body part of the email message normalized to the length of 1000 characters; The *header* subset is the header section of the email message normalized to the length of 100 characters; The *All* subset concludes the *From* field and the *Subject* field of the header section plus the whole body of the email massage. Also, every instance of *All* subset is normalized to the length of 1200 characters.

**Table 1.** The results from email classification using each kernel function with an comparison to [20]

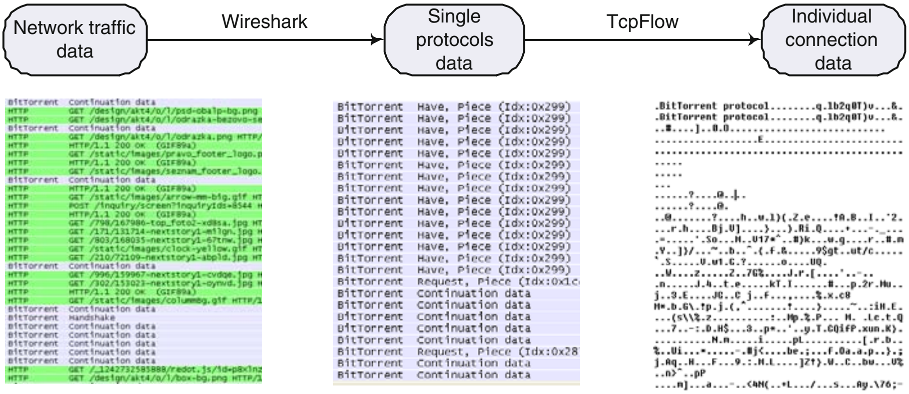
| Features | String Kernel Function | | | | Ref. Acc. [20] |
|---|---|---|---|---|---|
| | N-gram | Subsequence | Edit Distance | Bag of word | |
| Subject | 96.38 | **96.64** | 95.78 | 81.16 | 92.64 |
| Body | **99.28** | 99.23 | 97.19 | 81.04 | 84.41 |
| Header | **99.80** | **99.80** | 99.75 | 82.68 | 92.12 |
| All | 99.34 | **99.38** | 98.02 | 81.24 | 90.13 |

Table 1 presents the percentage accuracy of correctly classified email messages for each type of string kernel and email subset, where percentage accuracy of [20] is presented in the Ref. Acc. column for comparison. As seen from the table, the classification results reached by string kernel SVM are exceeding the percentage accuracy from the the reference paper [20]. The results for subset *Header* demonstrate that the first 100 characters of a message header is enough for correct spam classification. Results for other subsets are consistently good, however they are seemed to more susceptible to spammers tricks.

Among 4 string kernels, the outstanding performance of N-Gram and Subsequence kernel functions proves that the classification with substring/subsequence kernels is more suitable for spam detection than the kernels using whole string, such as Edit Distance. The Bag of word kernel is seen problematical on spam detection, this could be explained that these spammers normally use the same words for both spam and ham similarity evaluation.

### 4.2 Network Application Authentication

In this experiment, we used data from tcp network traffic produced by common network applications using different communication protocols like http, https, imap, pop3, ssh, ftp and some applications using dynamic ports like Bittorent. All network data was captured, and sorted by program *Wireshark* [15] into separated files according to protocols and then split into individual flows (connections) by program *tcpflow* [16] see Fig. 1.



```
BitTorrent  Continuation data
HTTP        GET /design/akt4/o/l/psd-obalp-bg.png
HTTP        GET /design/akt4/o/l/odrazka-bezovo-se
BitTorrent  Continuation data
HTTP        GET /design/akt4/o/l/odrazka.png HTTP/
HTTP        HTTP/1.1 200 OK  (GIF89a)
HTTP        GET /static/images/pravo_footer_logo.p
HTTP        HTTP/1.1 200 OK  (GIF89a)
HTTP        GET /static/images/seznam_footer_logo.
BitTorrent  Continuation data
HTTP        HTTP/1.1 200 OK  (GIF89a)
HTTP        GET /static/images/arrow-mm-big.gif HT
HTTP        POST /inquiry/screen?inquiryIds=8544 H
HTTP        HTTP/1.1 200 OK  (GIF89a)
HTTP        GET /798/167986-top_foto2-xd8sa.jpg HT
HTTP        GET /171/131714-nextstory1-m1lgn.jpg H
HTTP        GET /803/168035-nextstory1-67tnw.jpg H
HTTP        GET /static/images/clock-yellow.gif HT
HTTP        GET /210/72109-nextstory1-abpld.jpg HT
BitTorrent  Continuation data
HTTP        GET /996/159967-nextstory1-cvdqe.jpg H
HTTP        GET /302/153023-nextstory1-oynvd.jpg H
HTTP        HTTP/1.1 200 OK  (GIF89a)
HTTP        GET /static/images/colummbg.gif HTTP/1
BitTorrent  Continuation data
BitTorrent  Handshake
BitTorrent  Continuation data
BitTorrent  Continuation data
BitTorrent  Continuation data
BitTorrent  Continuation data
BitTorrent  Continuation data
HTTP        GET /_1242732585888/redot.js/id=p8x1nz
HTTP        GET /design/akt4/o/l/box-bg.png HTTP/1
```

```
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Have, Piece (Idx:0x299)
BitTorrent  Request, Piece (Idx:0x1c
BitTorrent  Continuation data
BitTorrent  Continuation data
BitTorrent  Continuation data
BitTorrent  Continuation data
BitTorrent  Continuation data
BitTorrent  Request, Piece (Idx:0x28
BitTorrent  Continuation data
BitTorrent  Continuation data
```

```
.BitTorrent protocol........q.lb2q0T)v...&.
.BitTorrent protocol........q.lb2q0T)v...&.
..#....]..0.D..............................
..................E........................
...........................................
.....
.....
...
......?....@..|..
......?....@.
..@........?....h..u.1)(.Z.e....!A.B..I..'2.
...r.h....Bj.U]....}...}.Ri.Q....+...-._...
.=.....'.So...M..U17*^..#)k...u.g....r..#.m
.Y..]}/...~..b..^.(.F.&.....9$gt..ut/c.....
`.S.....U.u1.C.?......o....UQ.
..W....z.....Z..7G%.....J.r.[....'..-..
.n.....J.4..t.e.....kT.I......#...p.2r.Hu..
j..3.E....JC..C j..F........%.x.c8
H*.b.G\.!p.j.(.^........!....}.....~..:iH.E.
...(s\\%.z...........:..Mp.%.P... M. .Lc.t.Q
...7..-:.D.H$...3..p*..'..y.T.CQifP.xun.K}.
...........N.n.....i.....pL..........[.r.b..
%..Ui...*.....-.#j<....be.;...F.Oa.a.p..}.;
j.Aq..H...F...9.:.M.L....]Z†}.W..C..bw...U%
..n>^..pP
....m]...a...-..<4M(..*L.../...s...Ay.\76;-
```

**Fig. 1.** Schema of data preparation for network application authentication. Network traffic data are sorted into separated files according to protocol with program *Wireshark* [15] and then split into individual flows using program *tcpflow* [16].

At preprocessing stage, we removed the traffic data in unreliable protocols like UDP, and reorder traffic data by the type of connections using program *tcpflow* [16]. All connections have been shorten to the length of 450 bytes and 750 bytes respectively, where connections shorter than 450/750 bytes are normalized by repeating its content. Every connection is labelled as its connection port number within the interval of $\langle 1, 49151 \rangle$. For connections using dynamic ports or the same port as other applications, we labelled the connections with some unoccupied ports. For example, Http, Msnms and Ocsp connections are noticed using the same port 80. To avoid repeat, we label Msnms and Ocsp as port 6 and 8, respectively. The summary of all connections is given in Table 2.

Table 3 presents the best percentage accuracies obtained by 4 types of string kernel. Percentage *Gen. Acc.* is recorded as a ratio of the correctly classified connection to total connection. Parameters *Lambda* and *C* have been gained from previous 5-cross validation tests. Parameter *Substring length* represents the length of contiguous (non-contiguous) substring for N-gram (Subsequence) kernel function.

**Table 2.** Number of connections (flows) for each protocol for training and testing

| Protocol name (port number) | Class label | Training set | Testing set |
|---|---|---|---|
| Http (80) | 80 | 675 | 235 |
| BitTorrent (dynamic) | 4 | 299 | 99 |
| Https (443) | 443 | 292 | 90 |
| Imap (993) | 993 | 42 | 14 |
| Pop3 (110) | 993 | 32 | 10 |
| Aol (5190) | 5190 | 21 | 7 |
| Ftp (21) | 21 | 12 | 4 |
| Msnms (80) | 6 | 9 | 4 |
| Microsoft-ds (445) | 445 | 9 | 3 |
| Ocsp (80) | 8 | 8 | 3 |
| Ssh (22) | 22 | 6 | 2 |
| Pptp (1723) | 1723 | 2 | 1 |
| Sum | | 1407 | 472 |

**Table 3.** The best results of network application classification for each kernel function

| Kernel Function | Parameters | | | | Gen. Acc |
|---|---|---|---|---|---|
| | Substring length | Connection length | C of C-SVC | Lambda | |
| Subsequence kernel | 4 | 750 | 65536 | 0.5 | 99.58% |
| N-gram kernel | 4 | 450 | 4096 | 0.5 | 99.58% |
| Edit distance kernel | - | 450 | 16 | 0.001 | 99.15% |
| Bag of word kernel | - | 750 | 1 | 0.25 | 49.75% |

As seen, *Subsequence* and *N-gram* kernel functions give the best 99.58% general accuracy, which follows that only 2 of total 472 connections are misclassified (two *Msnms* connections are misclassified as *Http* connections). It is worth noting that N-gram kernel function performs extremely well for distinguishing those network applications with shorter (450 chars) connection instance. However, the *Bag of word* kernel function is unable to recognize network applications because this kernel distinguishes network applications based on word similarity, when the word (continuous sequence of characters between any two gaps) represents a whole network connection, it is often too long for the kernel to differentiate applications.

Fig. 2 discloses the relationship between the substring length and general accuracy over network flows (connections) with length 450 and 750 chars, respectively. As seen, the general accuracy decreases over the length of substring for both N-gram and Subsequence kernels. This suggests that an longer substring normally leads to a decreased classification performance from the presented two string kernel functions.

In general, results in Table 3 and Fig. 2 have proved the effectiveness of applying Edit Distance, Subsequence and N-gram kernel functions for recognizing hidden network traffic including encrypted connections. Edit Distance kernel

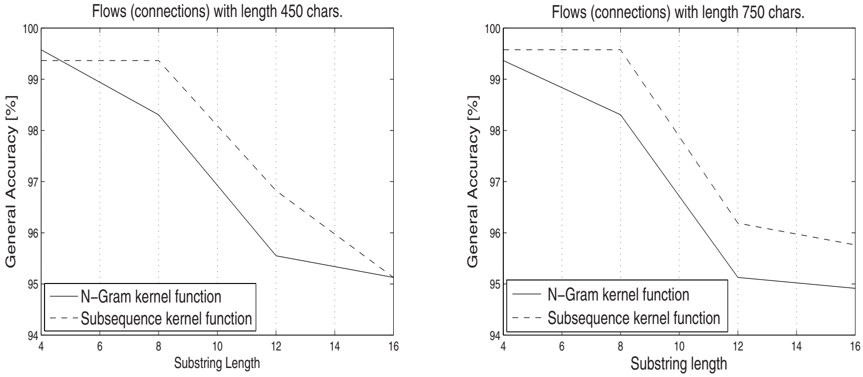


**Fig. 2.** Relation between the substring length and general accuracy for network flows (connections) with length 450 and 750 chars, respectively

identifies a global similarity of the string (connection), but it surprisingly gives a 99.15% authentication accuracy. This implies that functions based on complete string comparison are also suitable for network application recognition to some extent.

## 5 Conclusions and Future Work

In this paper, we propose a new network security technique for spam and hidden network application detection. Our technique is based on known string kernel functions with precomputed kernel matrices. In our implementation of the proposed technique, we used support vector machines with optimal configuration associated for each kernel function.

This paper makes two major contributions. First, a study of Spam detection. Our email classification results presents excellent ability of string kernels SVM to detect spam from relevant emails. Kernel functions using substrings/ subsequences are evidently less susceptible to spammers tricks than function which comparing the whole strings. As seen, the best classification accuracy has been reached for *Header* email subset and this finding proves the email header has critical influence on the spam classification. Second, a detection of hidden network application traffic. Our experimental analysis shows that sizes of first 450B of TCP connection are capable for accurate distinguishing network applications. From results is evident the suitable functions for application recognition from TCP connections are N-Gram, Subsequence and Edit Distance function. Our experiments prove the best results in network application recognition was gained with short length of substrings in function parameter.

For future work, we will continue to develop new string kernels, and address specially next network security tasks including testing memory dump and network intrusion detection.

## References

1. Chang, C.-C., Lin, C.-J.: LIBSVM:a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
2. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York (2004)
3. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Hoboken (2000)
4. Charras, C., Lecroqk, T.: Sequence comparison (1998), http://www-igm.univ-mlv.fr/~lecroq/seqcomp/index.html
5. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. J. Mach. Learn. Res. 2, 419–444
6. Fisk, M., Varghese, G.: Applying Fast String Matching to Intrusion Detection (September 2002)
7. Aizerman, A., Braverman, E.M., Rozoner, L.I.: Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control 25, 821–837 (1964)
8. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: COLT 1992: Proceedings of the fifth annual workshop on Computational learning theory, pp. 144–152. ACM, New York (1992)
9. Yuan, G.-X., Chang, C.-C., Lin, C.-J.: LIBSVM: libsvm experimental code for string inputs, http://140.112.30.28/~cjlin/libsvmtools/string/libsvm-2.88-string.zip
10. Scarfone, K., Mell, P.: Guide to intrusion detection and prevention systems (idps). In: NIST: National Institute of Standards and Technology (2007), http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf
11. Vapnik, V.N.: The nature of statistical learning. Springer, New York (1995)
12. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
13. Caswell, B., Beale, J., Foster, J.C., Faircloth, J.: Snort 2.0 Intrusion Detection. Syngress (2003), http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&amp;path=ASIN/1931836744
14. Whitman, M.E., Mattord, H.J.: Principles of Information Security. Course Technology Press, Boston (2004)
15. Combs, G., et al.: Wireshark: network protocol analyzer, http://www.wireshark.org/
16. Elson, J.: tcpflow: tcpflow reconstructs the actual data streams and stores each flow in a separate file for later analysis, http://www.circlemud.org/jelson/software/tcpflow/
17. Bogomolny, A.: Distance Between Strings, http://www.cut-the-knot.org/doyouknow/Strings.shtml
18. SpamAssassin public mail corpus, http://spamassassin.apache.org/publiccorpus/
19. Spam dataset, http://www.em.ca/7Ebruceg/spam/
20. Lai, C.-C.: An empirical study of three machine learning methods for spam filtering. Knowledge-Based Systems 20, 249–254 (2007)

# Automated Log Analysis of Infected Windows OS Using Mechanized Reasoning

Ruo Ando

National Institute of Information and Communication Technology,
4-2-1 Nukui-Kitamachi, Koganei,
Tokyo 184-8795 Japan
ruo@sfc.keio.ac.jp

**Abstract.** Malware (Malicious Software) of Windows OS has become more sophisticated. To take some countermeasures for recent infection, more intelligent and automated system log analysis is necessary. In this paper we propose an automated log analysis of infected Windows OS using mechanized reasoning. We apply automated deduction system for gathering events of malware and extract the behavior of infection over large scale system logs. In experiment, we cope with four kinds of resolution strategies to detect the malicious behavior. It is shown that automation of analyzing system logs is possible for detecting actual malicious software.

**Keywords:** Automated log analysis, malicious software, incident detection, Windows OS, mechanized reasoning.

## 1 Introduction

Cyber attack has become more sophisticated. Malware (Malicious Software) has been more complicated for more stealth and well-organized behavior. To cope with this problem, more intelligent and automated analysis is necessary. In this paper we propose the automation of log analysis of infected Windows OS using mechanized reasoning.

### 1.1 Detecting Infection of Windows OS

Malware of Windows OS has become more well-organized. The malicious behavior has been more complicated and hidden in system logs in more crafted way. The behavior is split over the large scale system log, which impose a great burden of security experts to detect the infection and discover the evidence. to take some countermeasures of this situation, automation of detecting infection of Windows OS is necessary.

### 1.2 Large Scale Log Analysis

Recent malware such as BotNet and Rootkit are more stealth by acting like normal process or activating deeper under the system. As result, if we try to

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 540–547, 2009.

detect these, large scope of system logs are necessary. We need to gather evidence of events over system logs to trace the state transition of Windows OS infected by mlawares. Large scale log analysis is required to extract recent and sophisticated malwares such as BotNet and Rootkit.

## 2 Mechanized Reasoning

Mechanized reasoning is also called as automated reasoning in which fields researchers cope with the creation of software which makes computers ”reason” in the sense of mathematical aspects such as solving puzzle and proving theorems. In this field, software such as FoL (first-order logic) or HoL (higher order logic) theorem prover, SAT solver and model generator is created to automated the mathematical process. In this paper we apply automated deduction system for automated log analysis of infected Windows OS using mechanized reasoning.

### 2.1 Set of Support Strategy

Set of support was introduced by L.Wos, S.Robinson and Carson in 1965[1]. If the clause T is retrieved from S, SOS is possible with the satisfiability of S-T. Set of support strategy enable the researcher to select one clause characterizing the searching to be placed in the initializing list called SOS. For the searching to be feasible and more effective, the resolution of more than one clauses not in SOS is inhibited in order to prevent the prover go into abundant searching place. Figure1. Set of Support resolution strategy Figure1 show the resolution process in set of support strategy, where S=P and Q and R, P and R, Q and R, R. The restriction imposes the reasoning so that the program do not apply an inference rule to a set of clauses that are not the complement of set of support.

### 2.2 Hyperresolution

In generating encoder, we apply the inference rule called hyper resolution[2], which is a kind of resolution that can do resolutions at once compared with several steps in another rules. For hyperresolution, these must be the negative or mixed clause with the remaining clauses equal to the number of literals in the negative or mixed clause. The positive clause are described as satellites, the negative clause nucleus. ”Hyper” means that in this resolution more process has occur than another resolution such as binary resolution.

### 2.3 Weighting Strategy

In contrast to the restriction strategies which block a reasoning path to apply an inference rules to many kinds of subsets of clauses, a direction (ordering) strategy dictates what to set focus on next. Among these strategies, we apply weighting strategy[3]. With this direction strategy, we can assign priorities to terms, clauses and concepts which make it possible to reflect the knowledge and intuition about

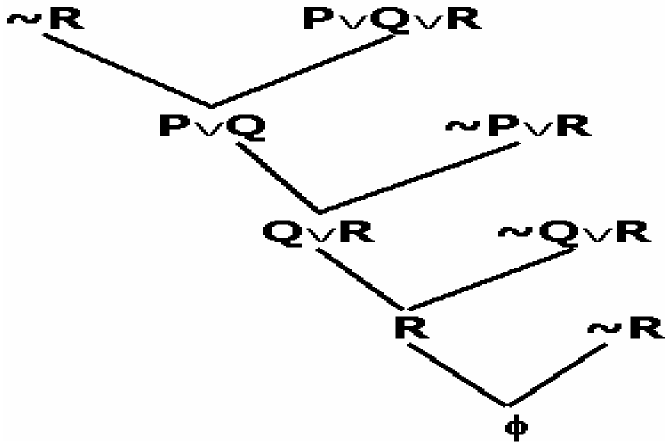


**Fig. 1.** Figure1 show the resolution process in set of support strategy, where S={P and Q and R, P and R, Q and R, R}.The restriction imposes the reasoning so that the program do not apply an inference rule to a set of clauses that are not the complement of set of support.

how a reasoning program should proceed. Weighting strategy, opposite to the restriction strategy called Set of Support, can be used complementarily. In the situation where the set of support strategy is adopted, the remaining clauses can be processed to complete the application of various inference rules. After the clauses assigned to set of support have been focus of attention, we can choose among the new clauses generated and retained for the following various kind of inference application. With weighting, you can supply various criteria for defining interests and for defining complex information. After the input clauses have been focus of attention, the choice is made from among the new clauses . those clauses that have been generated and retained because they represent new information that is deemed acceptable.

### 2.4 Subsumption

Subsumption[4] is the process of discarding a specific statement. The clause that duplicated or is less general is discarded in the already-existing information. As a result, subsumption prevents a reasoning program from retaining clauses that is obviously redundant, especially is logically captured by more general clauses. For example,

```
OLDER(father(x),x)
Subsumes
OLDER(father(Ann),Ann).
```

**Definition.** The clause A subsumes the clause B when B is the instance that is logically captured by B.

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

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

## 3 Logging Techniques of Windows OS

Windows modification OS consists of three steps: [1]inserting DLL into user process, [2]inserting filter driver into kernel space. In this section we discuss library insertion (DLL injection) and filter driver injection.

### 3.1 DLL Injection

DLL injection is the insertion of original library on userland. We apply DLL injection for inspecting illegal resource access of malicious process. DLL injection is debugging technology to hook API call of target process. Windows executable applies some functions from DLL such as kernel32. dll. Executable has import table to use the linked DLL. This table is called as import section. Among some techniques of DLL injection, modifying import table is useful because this technique is CPU-architecture independent. Figure 5 show the modification of import table. Address of function A on left side is changed to the address of inserted function on right side. In code table, some original functions are appended to executable. Modified address is pointed to code of inserted function. By doing this, when the function A is invoked, the inserted function is executed.

### 3.2 Filter Driver

Filter driver is an intermediate driver which runs between kernel and device driver. By using filter driver, we can hook events on lower level compared with library insertion technique on userland. In detail, System call table is modified to insert additional routine for original native API. In proposed system, filter driver is implemented and inserted for hooking events on file system.

## 4 Experimental Result

In this section we show the experimental result of analyzing log of Windows OS infected by malware by Sassar-A. Sassar-A is one of the most famous malicious software which exploits a vulnerability of lsass.exe (Windows daemon process). Once Sassar succeeded to infect Windows OS, it reproduce itself into Windows

**Table 1.** Malicious action of Sassar-A

| action | type | objective |
|---|---|---|
| Execute | Passive | User |
| Infect | Reproduction | SVCHOST. |
| Infect | Installation | Windows Directory. |
| Daemonize | Resident process | Memory. |
| Infect | Exploitation | Networked computer. |

**Table 2.** Sample APIs to be hooked in proposed system

| API name | API type A | API type B |
|---|---|---|
| AllocVirtualMemory | Native API | Memory allocation. |
| CreateSection | Native API | Memory section creation. |
| LoadLibrary | User API | Invoking library. |
| MapViewOfSection | Native API | Memory Allocation. |
| PreRead | filter function | File Access. |
| PreWrite | filter function | File Access. |
| ZwCreateEvent | Native API | Memory Allocation. |
| ZwOpenKeys | Native API | Register Access. |
| CreateProcess | User API | Process Operation. |
| ZwOpenKeys | Native API | Registry Access. |

**Table 3.** Number of empty clauses in depth or width search in two types of resolution

| weight | hyper-stack | hyper-queue | binary-stack | binary-queue |
|---|---|---|---|---|
| 20 | 1 | 1 | 2 | 2 |
| 19 | 1 | 1 | 2 | 2 |
| 15 | 1 | 1 | 2 | 2 |
| 14 | 1 | 1 | 2 | 2 |
| 13 | 1 | 1 | 2 | 1 |
| 12 | 1 | 1 | 2 | 0 |
| 11 | 1 | 1 | 2 | 0 |
| 10 | 1 | 1 | 0 | 0 |
| 5 | 1 | 1 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |

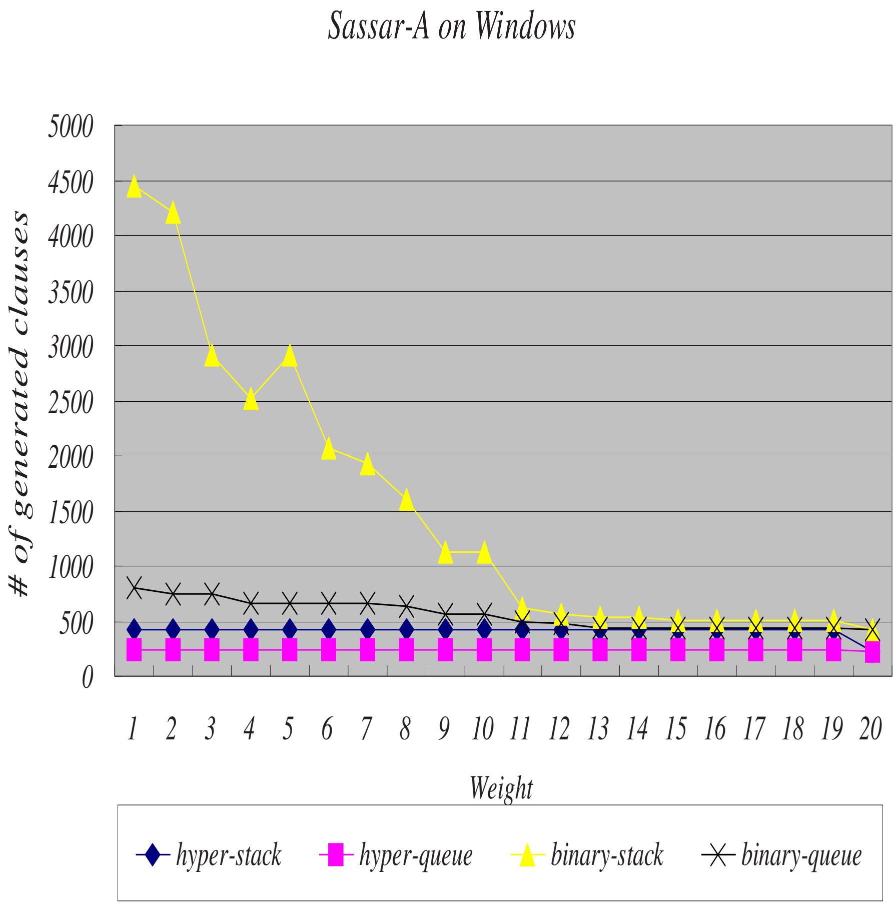


**Fig. 2.** The number of generated clauses in extracting the bahavior of Sassar-A. There are four kinds of resolution strategies: hyper-stack, hyper-queue, binary-stack and binary-queue. Stack means breadth-first search and queue means width-first search. Hyper-queue (hyperresolution with breadth-first search) is the most effective while birary-stack (binary resolution with breadth-first search) is the most weight sensitive. Hyper resolution is more stable than binary resolution with the threshold in missing the infection point.

directory with name SVSHOST.exe. And then, it becomes daemon process as avserve. Also, it tries to infect networked computers. Table 1 shows the action of Sassar-A on the infection of Windows OS. To detect the behavior of Sassar-A, these actions needs to be formulated to process automated deduction system.

To analyze the system log and extract the behavior of malware such as Sassar-A, APIs shown in Table 2 are intercepted to reveal the actions below the application layer of Windows OS. Table 2 shows the example of APIs in three kinds: Native API, User API and filter function. On the view of system behavior, there are memory allocation, memory section creation, invoking library, file access, register access and process operation (creation).

Table 3 shows the detection result with hyper and binary resolution in width an breadth first search. In around weight 10 - 13, automated deduction system misses the infection point in both resolution strategies. Figure 2 shows the calculation cost (number of generated clauses) in four strategies. Hyper-queue (hyperresolution with breadth-first search) is the most effective while birary-stack (binary resolution with breadth-first serach) is the most weight sensitive. Hyper resolution is more stable than binary resolution with the threshold in missing the infection point.

## 5 Conclusions

With the rapid advances of system and application programming technologies, malware (Malicious Software) of Windows OS has become more sophisticated. To take some countermeasures for recent infection, more intelligent and automated system log analysis is necessary. In this paper we have proposed the automated log analysis of infected Windows OS using mechanized reasoning. We have applied automated deduction system for gathering events of malware and extract the behavior of infection over large scale system logs. In experiment, we have coped with four kinds of resolution strategies to detect the malicious behavior. Access of system resources such as memory, file, registry is intercepted by API hooking. We have shown generated clauses in four resolution strategies: hyper-stack, hyper-queue, binary-stack and binary-queue. It is shown that there are optimum weight around 10 to detect malicious behavior with the least calculation cost. We can conclude that automation of analyzing system logs is possible for detecting actual malicious software.

## References

1. Wos, L., Robinson, G.A., Carsonm, D.F.: Efficiency and Completeness of the Set of Support Strategy in Theorem Proving. Journal of Automated Reasoning (1965)
2. Wos, L.: The Problem of Explaining the Disparate Performance of Hyperresolution and Paramodulation. J. Autom. Reasoning 4(2), 215–217 (1988)
3. Wos, L.: The Problem of Self-Analytically Choosing the Weights. J. Autom. Reasoning 4(4), 463–464 (1988)
4. Wos, L.: The Problem of Choosing the Type of Subsumption to Use. J. Autom. Reasoning 7(3), 435–438 (1991)
5. Wos, L., Robinson, G.A., Carson, D.F., Shalla, L.: The Concept of Demodulation in Theorem Proving. Journal of Automated Reasoning (1967)
6. McCune, W.: 33 basic test problems: A practical evaluation of some paramodulation strategies. Preprint ANL/MCS-P618-1096, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL (1996)
7. Wos, L., Pieper, G.W.: The Hot List Strategy. Journal of Automated Reasoning (1999)
8. OTTER automated deduction system, http://www.mcs.anl.gov/AR/otter/

9. McCune, W.: OTTER 3.3 Reference Manual CoRR cs.SC/0310056 (2003)
10. Wos, L.: The Power of Combining Resonance with Heat. Journal of Automated Reasoning (1996)
11. Spinellis, D.: Reliable identification of bounded-length viruses is NP-complete. IEEE Transactions on Information Theory (2000)
12. Ando, R., Takefuji, Y.: Faster resolution based metamorphic virus detection using ATP control strategy. Wseas Transactions on Information Science And Applications 3(2), 260–2266 (2006)
13. Ando, R.: Faster parameter detection of polymorphic viral code using hot list strategy. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008, Part I. LNCS, vol. 5506, pp. 555–562. Springer, Heidelberg (2008)
14. Ando, R.: Parallel analysis of polymorphic viral code using automated deduction system. In: 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007), Qingdao, China (July 2007)

# HumanBoost: Utilization of Users' Past Trust Decision for Identifying Fraudulent Websites

Daisuke Miyamoto[1], Hiroaki Hazeyama[2], and Youki Kadobayashi[2]

[1] National Institute of Information and Communications Technology
Traceable Network Group
4-2-2 Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan
daisu-mi@nict.go.jp

[2] Nara Institute of Science and Technology
Graduate School of Information Science
Internet Engineering Laboratory
8916-5 Takayama,
Ikoma, Nara 630-0192, Japan
hiroa-ha@is.naist.jp,
youki-k@is.naist.jp

**Abstract.** In this paper, we present an approach that aims to study users' past trust decisions (PTDs) for improving the accuracy of detecting phishing sites. Generally, Web users required to make trust decisions whenever their personal information is asked for by websites. We assume that the database of users' PTDs would be transformed into a binary vector, representing phishing or not, and the binary vector can be used for detecting phishing sites similar to the existing heuristics. For our pilot study, we invited 10 participants and performed a subject experiment in November 2007. The participants browsed 14 emulated phishing sites and 6 legitimate sites, and checked whether the site appeared to be a phishing site or not. By utilizing participants' trust decision as a new heuristic, we let AdaBoost incorporate the heuristic into 8 existing heuristics. The results show that the average error rate in the case of HumanBoost is 9.5%, whereas that in the case of participants is 19.0% and that in the case of AdaBoost is 20.0%. Thus, we conclude that HumanBoost has a potential to improve the detection accuracy for each Web user.

## 1 Introduction

Phishing is a form of identity theft in which the targets are users rather than computer systems. A phishing attacker attracts victims to a spoofed website, a so-called "phishing site", and attempts to persuade them to provide their personal information. The damage suffered from phishing is increasing. In 2005, the Gartner Survey reported that 1.2 million consumers lost 929 million dollars as a result of phishing attacks [1]. The modern survey conducted in 2007 also reported that 3.6 million consumers lost 3 billion dollars [2].

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 548–555, 2009.

To deal with phishing attacks, a heuristics-based detection method [3] has begun to garner attention. A heuristic is an algorithm to distinguish phishing sites from the others based on users' experience, and a heuristic checks if a site appears to be a phishing site. For example, checking the life time of the issued website is one of the famous heuristics; the most of the phishing sites' URL are expired in short time period [4]. On the basis of the detection result from each heuristic, the heuristic-based solution calculates the likelihood of a site being a phishing site and compares the likelihood with the defined discrimination threshold. Unfortunately, the detection accuracy of existing heuristic-based solutions is far from suitable for practical use [5], even if various studies discovered heuristics.

In our previous work [6], we attempted to improve the accuracy by employing machine learning techniques for combining heuristics, since we assumed the inaccuracy is caused by heuristics-based solutions that can not use these heuristics appropriately. We evaluated the performance of machine learning-based detection methods (MLBDMs). MLBDMs must achieve high detection accuracy, and they must adjust their detection strategies for Web users, so that the performance metrics consisted of $f_1$ measure, error rate, and Area Under the ROC Curve (AUC). Our evaluation results showed that the highest $f_1$ measure was 0.8581, the lowest error rate was 14.15%, and the highest AUC was 0.9342, all of which were observed in the case of the AdaBoost-based detection method. For the most of the cases, MLBDMs performed better than the existing detection method.

In this paper, we propose "HumanBoost", which aims improving the AdaBoost-based detection methods. The key idea of HumanBoost is utilizing Web users' past trust decisions (PTDs). Basically, human beings have the potential to identify phishing sites, even if the existing heuristics cannot detect them. If we could construct the database of PTD for each Web user, it would be able to use the record of the user's trust decision as one feature vector on detecting phishing sites. HumanBoost also involves the idea of adjusting the detection for each Web user. If a user is a security expert, the most predominant factor on detecting phishing sites would be his/her trust decision. Conversely, the existing heuristic will have a strong effect on detection when the user is a novice and his/her PTD has often failed.

As our pilot study, we invited 10 participants and performed a subject experiment. The participants browsed 14 emulated phishing sites and 6 legitimate sites, and checked whether the site appeared to be a phishing site or not. By utilizing participants' trust decision as a new heuristic, we let AdaBoost incorporate the heuristic into 8 existing heuristics. The results show that the average error rate in the case of HumanBoost was 9.5%, whereas that in the case of participants was 19.0% and that in the case of AdaBoost was 20.0%.

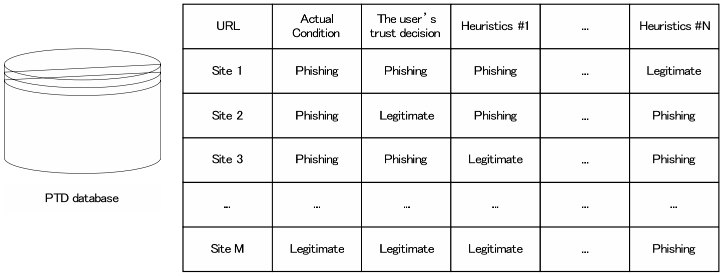
The rest of this paper is organized as follows. In Section 2, we summarize the related work, and explain our proposal in Section 3. In Section 4, we describe our preliminary evaluation, and discuss the limitation of HumanBoost in Section 5. Finally, we conclude our contribution in Section 6.

## 2 Related Work

For mitigating phishing attacks, machine learning, which facilitates the development of algorithms or techniques by enabling computer systems to learn, has begun to garner attention. PFILTER, which was proposed by Fette et al. [7], employed SVM to distinguish phishing emails from other emails. Abu-Nimeh et al. compared the predictive accuracy of 6 machine learning methods [8]. They analyzed 1,117 phishing emails and 1,718 legitimate emails with 43 features for distinguishing phishing emails. Their research showed that the lowest error rate was 7.72% in the case of Random Forests. Ram Basnet et al. performed an evaluation of 6 different machine learning-based detection methods [9]. They analyzed 973 phishing emails and 3,027 legitimate emails with 12 features, and showed that the lowest error rate was 2.01%. The experimental conditions were different between [8] and [9]; however, machine learning provided high accuracy for the detection of phishing emails.

Apart from phishing emails, machine learning was also used to detect phishing sites. Pan et al. presented an SVM-based page classifier for detecting those websites [10]. They analyzed 279 phishing sites and 100 legitimate sites with 8 features, and the results showed that the average error rate was 16%.

Our previous work employed 9 machine learning techniques [6]. We employed 8 heuristics presented in [11] and analyzed 3,000 URLs, which consisted of 1,500 legitimate sites and the same number of phishing sites, reported on PhishTank.com from November, 2007 to February, 2008. Our evaluation results showed that the highest $f_1$ measure was 0.8581, the lowest error rate was 14.15%, and the highest AUC was 0.9342, all of which were observed in the case of the AdaBoost-based detection method. In the most of the cases, MLBDMs performed better than the existing detection method.



| URL | Actual Condition | The user's trust decision | Heuristics #1 | ... | Heuristics #N |
|---|---|---|---|---|---|
| Site 1 | Phishing | Phishing | Phishing | ... | Legitimate |
| Site 2 | Phishing | Legitimate | Phishing | ... | Phishing |
| Site 3 | Phishing | Phishing | Legitimate | ... | Phishing |
| ... | ... | ... | ... | ... | ... |
| Site M | Legitimate | Legitimate | Legitimate | ... | Phishing |

**Fig. 1.** PTD database

## 3 HumanBoost

The key idea of HumanBoost is utilizing Web users' past trust decisions (PTDs). Generally, Web users are required to make trust decisions whenever they input their personal information into websites. In other words, we assumed that a Web user outputs binary variable, phishing or legitimate, when the website requires users to input their password. It is similar to the existing heuristics.

In HumanBoost, we assume that each Web user has his/her own PTD database, as shown in Fig. 1. The schema of the PTD database consists of the website's URL, actual conditions, the result of the user's trust decision, and the results from existing heuristics. It is to be noted that we do not propose to share the PTD database among users because of privacy concerns.

The PTD database can be regarded as a training dataset that consisted of $N + 1$ binary explanatory variables and 1 binary response variable. Hence, we employ a machine learning technique for studying this binary vector for each user's PTD database. In this study, we employ the AdaBoost algorithm because AdaBoost performed better in our previous work [6].

Further, we expect AdaBoost to cover each user's weak points. Essentially, the boosting algorithms assign high weight to a classifier that correctly labels a site where other classifiers had labeled incorrectly. Assuming that a user's trust decision can be treated as a classifier. AdaBoost would cover users' weak points by assigning high weights on heuristics that can correctly judge the site where he/she is likely to misjudge.

## 4 Experiment and Results

As our pilot study, in November 2007, we performed a subject experiment by using legitimate enterprise websites and emulated phishing sites. We invited 10 participants who belonged to Nara Institute of Science and Technology, all of them were male, 3 of them completed their M.Eng degree in the last 5 years, while the remaining were master's degree students.

In Section 4.1, we describe how we constructed the dataset, and we explain the design of our experiments in Section 4.2. Finally, we show the results of our preliminary experiment in Section 4.3.

### 4.1 Dataset Description

Similar to the typical phishing IQ tests performed by Dhamija et al. [12], we prepared 14 emulated phishing sites and 6 legitimate ones, all of which contained Web input forms on which users could input their personal information such as user ID and password. The conditions of the sites are shown in Table 1. Some phishing sites are derived from actual phishing sites according to a report from Phishtank.com. Other phishing sites are emulated phishing sites that mimic actual websites by using phishing subterfuges [7,13,14] to induce participants to input their personal information.

**Table 1.** Conditions in each site

| # | Website | Real / Spoof | Lang | Description |
|---|---|---|---|---|
| 1 | Live.com | real | EN | URL (login.live.com) |
| 2 | Tokyo-Mitsubishi UFJ | spoof | JP | URL(www-bk-mufg.jp) |
| 3 | PayPal | spoof | EN | URL (www.paypal.com.%73%69 ... %6f%6d) (URL encoding abuse) |
| 4 | Goldman Sachs | real | EN | URL(webid2.gs.com), SSL |
| 5 | Natwest Bank | spoof | EN | URL(onlinesession-0815.natwest.com.esb6eyond.gz.cn) (Derived from PhishTank.com) |
| 6 | Bank of the West | spoof | EN | URL (www.bankofthevvest.com) |
| 7 | Nanto Bank | real | JP | 3rd party URL (www2.answer.or.jp), SSL |
| 8 | Bank of America | spoof | EN | URL( bankofamerica.com@index.jsp-login-page.com ) (URL scheme abuse) |
| 9 | PayPal | spoof | EN | URL (www.paypal.com) but first `a` letter is a Cyrillic small letter `a` (U+430) (IDN abuse) |
| 10 | Citibank | spoof | EN | URL(IP address) |
| 11 | Amazon | spoof | EN | URL (www.importen.se), contains `amazon` in its path (Derived from PhishTank.com) |
| 12 | Xanga | real | EN | URL (www.xanga.com) |
| 13 | Morgan Stanley | real | EN | URL (www.morganstanleyclientserv.com), SSL |
| 14 | Yahoo | spoof | EN | URL(IP address) |
| 15 | U.S.D of Treasury | spoof | EN | URL (www.tarekfayed.com) (Derived from PhishTank.com) |
| 16 | Sumitomo Mitsui Card | spoof | JP | URL (www.smcb-card.com) |
| 17 | eBay | spoof | EN | URL (secuirty.ebayonlineregist.com) |
| 18 | Citibank | spoof | EN | URL (VeCoN.com) ( is pronounced "Shi Tei Ban Ku", look-alike "CitiBank" in Japanese Letter) (IDN abuse) |
| 19 | Apple | real | EN | URL (connect.apple.com), SSL, popup warning by accessing non-SSL content |
| 20 | PayPal | spoof | EN | URL ( www.paypal.com@verisign-registered.com ) (URL scheme abuse) |

### 4.2 Design of Experiment

We used a within-subjects design, where every participant saw every website and judged whether the site was deemed a phishing site or not. In our test, we asked 10 participants to browse the websites freely. We installed Windows XP on each participant's system with Internet Explorer (IE) version 6.0 as the browser. Other than configuring IE to display IDN, we did not install security softwares and/or anti-phishing toolbars. We also did not prohibit participants to access websites that were not listed in Table 1. Therefore, some participants inputted several terms into search engines and compared the URL of the site with the URLs of those listed in Google's search result pages.

The detection results by each participant are shown in Table 2. In addition, "#" denotes the number of websites in Table 1, P1 - P10 denote the 10 participants, the letter "F" denotes that a participant failed to judge the website, and the empty block denotes that a participant succeeded in judging correctly.

### 4.3 Results of Experiment

At first, we determined the detection accuracy of the AdaBoost-based detection methods. We employed 8 heuristics, all of which were proposed by Zhang et al. [11] and outputted a binary variable representing phishing or not.

**Table 2.** The detection result by each participant

| # | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | F | F | | |
| 2 | F | F | | | | | | | | |
| 3 | | F | | | | | | | | |
| 4 | | | F | | | F | | | F | |
| 5 | | | | | | | F | F | | |
| 6 | | F | F | | F | F | | | | |
| 7 | F | F | | F | | F | | | | |
| 8 | | | | | | | | | | |
| 9 | | | | | | | | | | |
| 10 | | | | F | | | | | | |
| 11 | | | | | | | F | | | |
| 12 | | | | | | | F | | F | |
| 13 | | F | F | | | | | | | |
| 14 | | | | | | | | F | | |
| 15 | | | | F | | | | | | |
| 16 | | F | | F | | | | F | | |
| 17 | | F | | F | | | | F | | |
| 18 | | | | | | | | | | |
| 19 | F | | | F | | | F | F | F | |
| 20 | | | F | | | | | | | |

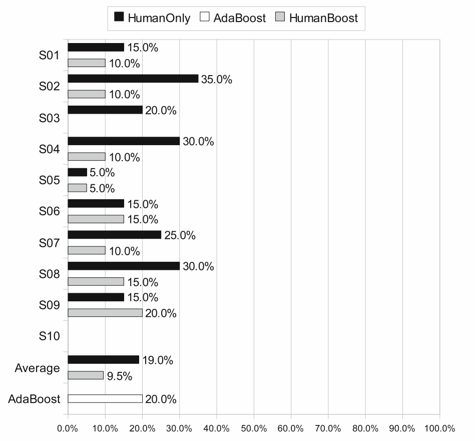


**Fig. 2.** Error Rate in HumanOnly, AdaBoost and HumanBoost

In this evaluation, we performed 4-fold cross validation to average the result. However, we considered that the experiment involved a small, homogeneous test population, and so it would be difficult to generalize the results to typical phishing victims. We will discuss our plan for removing the bias in Section 5. In addition, we selected 1 as the iteration time, and decided to use the average error rate as a performance metric. On the basis of this condition, we observed that the average error rate of the AdaBoost-based detection method was 20.0%.

Next, we also calculated the detection accuracy of our proposed HumanBoost. We constructed 10 of the PTD database. In other words, we made 10 types of $20 \times 9$ binary vectors. Under the same condition described above, we calculated the average error rate for each case of the HumanBoost.

The results were shown in Fig. 2, where HumanOnly denotes a detection accuracy without using AdaBoost and/or HumanBoost. By comparing the case of HumanBoost with the case of HumanOnly, the error rate was lesser or equal in the most of the cases. The average error rate in the case of HumanBoost was 9.5%, whereas the average error rate in the case of HumanOnly was 19.0% and that in the case of AdaBoost was 20.0%. In addition, we performed paired t-test and observed that there was a statistical difference between the accuracy in the case of HumanBoost and that in the case of HumanOnly.

In particular, the average error rate of P9 decreased from 85.0% (HumanOnly) to 80.0%(HumanBoost). We found that some heuristics were assigned higher weights than P9's trust decision. In our experiment, P9 had labeled 3 legitimate sites as phishing sites, whereas the existing heuristics had labeled these 3 sites correctly. Accordingly, the detection of P9 was overwhelmed by that of existing heuristics. We assumed that this is the reason for increasing the error rate.

## 5 Discussion

Basically, removing bias is important for a participant-based test. Although we used cross validation and paired t-test to eliminate bias, it still can be assumed that there was bias due to the number of samples and/or biased samples. We positioned our laboratory test as a first step, and decided to perform a field test in a large-scale manner. The one approach toward field test is implementing HumanBoost-capable phishing prevention system. It can be possible by distributing the work as browser-extension with some data collection and getting a large population of users to agree to use it.

The weak point of the HumanBoost-capable system is that the system always works after the user finished making the trust decision. Generally, phishing prevention systems will work for users to avoid visiting phishing sites. Apart from these systems, HumanBoost requires users to judge if their secret can be inputted into the site. To protect the users, the HumanBoost-capable system should cancel the input or the submission of users' secret.

Another problem is the difficulty in convincing users to reconsider their trust decision. When a user attempts to browse a phishing site, usual phishing prevention systems display some alert messages that he/she could be visiting a phishing site. In HumanBoost, such messages would be shown after making the trust decision. Otherwise the user recalls his/her trust decision, the HumanBoost-capable system would not block phishing attacks even if the system alerts correctly.

## 6 Conclusion

In this paper, we presented an approach called HumanBoost to improve the detection accuracy of phishing sites. The key idea was utilizing users' past trust decisions(PTDs). Since Web users might be required of making trust decisions whenever they input their personal information into websites, we considered to record these trust decisions for learning purposes. We simply assumed that the record can be described by a binary variable, representing phishing or not, and found that the record was similar to the output of the existing heuristics.

As our pilot study, we invited 10 participants and performed a subject experiment in November 2007. The participants browsed 14 emulated phishing sites and 6 legitimate sites, and checked whether the site appeared to be a phishing site or not. By utilizing participants' trust decision as a new heuristic, we let AdaBoost incorporate the heuristic into 8 existing heuristics. The results show that the average error rate in the case of HumanBoost was 9.5%, whereas that of participants was 19.0% and that in the case of AdaBoost was 20.0%. Our paired t-test showed that there was a statistical difference. Thus, we concluded that HumanBoost has a potential to improve the detection accuracy for each Web user.

We also discussed to perform our tests in lesser biased ways. To facilitate the field test in a large-scale manner, we mentioned the limitation of HumanBoost from the aspect of the system design. We found that HumanBoost-capable system should have an ability of canceling the input or the submission of users' secret, instead of blocking phishing sites. We also found that the HumanBoost-capable system should have some interfaces that can expedite users re-making trust decisions.

Apart from its development, we attempt to introduce fuzzy factors for users' trust decision. Users can suspend their trust decision instead of labeling a site as phishing or not. Essentially, machine learning techniques can manipulate quantitative variables, so that we do not adhere to categorical variables in our future work.

## References

1. McCall, T., Moss, R.: Gartner Survey Shows Frequent Data Security Lapses and Increased Cyber Attacks Damage Consumer Trust in Online Commerce (2005), http://www.gartner.com/press_releases/asset_129754_11.html
2. McCall, T.: Gartner Survey Shows Phishing Attacks Escalated in 2007; More than $3 Billion Lost to These Attacks (2007), http://www.gartner.com/it/page.jsp?id=565125
3. Chou, N., Ledesma, R., Teraguchi, Y., Boneh, D., Mitchell, J.C.: Client-side defense against web-based identity theft. In: Proceedings of 11th Annual Network and Distributed System Security Symposium (2004)
4. Anti-Phishing Working Group: Phishing Activity Trends Report - Q1 (2008), http://www.apwg.com/reports/apwg_report_Q1_2008.pdf
5. Zhang, Y., Egelman, S., Cranor, L., Hong, J.: Phinding Phish: Evaluating Anti-Phishing Tools. In: Proceedings of the 14th Annual Network and Distributed System Security Symposium (2007)
6. Miyamoto, D., Hazeyama, H., Kadobayashi, Y.: An Evaluation of Machine Learning-based Methods for Detection of Phishing Sites. Australian Journal of Intelligent Information Processing Systems 10(2), 54–63 (2008)
7. Fette, I., Sadeh, N.M., Tomasic, A.: Learning to detect phishing emails. In: Proceedings of the 16th International Conference on World Wide Web (2007)
8. Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A Comparison of Machine Learning Techniques for Phishing Detection. In: Proceedings of eCrime Researchers Summit (2007)
9. Basnet, R., Mukkamala, S., Sung, A.H.: Detection of Phishing Attacks: A Machine Learning Approach. Studies in Fuzziness and Soft Computing 226, 373–383 (2008)
10. Pan, Y., Ding, X.: Anomaly Based Web Phishing Page Detection. In: Proceedings of the 22nd Annual Computer Security Applications Conference on Annual Computer Security Applications Conference (2006)
11. Zhang, Y., Hong, J., Cranor, L.: CANTINA: A Content-Based Approach to Detect Phishing Web Sites. In: Proceedings of the 16th World Wide Web Conference (2007)
12. Dhamija, R., Tygar, J.D., Hearst, M.A.: Why Phishing Works. In: Proceedings of Conference on Human Factors in Computing Systems (2006)
13. Felten, E.W., Balfanz, D., Dean, D., Wallach, D.S.: Web Spoofing: An Internet Con Game. Technical Report 540-96 (Department of Computer Science, Princeton University)
14. Fu, A.Y., Deng, X., Wenyin, L., Little, G.: The methodology and an application to fight against Unicode attacks. In: Proceedings of the 2nd Symposium on Usable Privacy and Security (2006)

# A Methodology for Analyzing Overall Flow of Spam-Based Attacks

Jungsuk Song[1], Daisuke Inoue[1], Masashi Eto[1], Mio Suzuki[1],
Satoshi Hayashi[2], and Koji Nakao[1]

[1] National Institute of Information and Communications Technology
{song,dai,eto,mio,ko-nakao}@nict.go.jp
[2] Symantec Japan Research Institute
satoshi_hayashi@symantec.com

**Abstract.** Over the last decade, unsolicited bulk e-mails, i.e., spams, have been dramatically increasing and they have been definitely recognized as a serious Internet threat. Especially, recent spams mostly caused by various malwares (e.g., bots, worms) often contain URLs that navigate spam receivers to malicious Web servers for the purpose of malware infection. In addition, malwares such as bots operate in cooperation with each other, and there are close links between malwares and malicious Web servers. In this paper, considering the need for further studies on the mitigation of recent spam-based attacks, we propose a methodology for analyzing their overall flow in order to investigate the active relationship among spams, malwares and malicious Web servers. Furthermore, we have evaluated our method using double bounce e-mails obtained from our own SMTP server. The experimental results show that the proposed method is highly effective to analyze the correlation between spams' sources and their eventual destinations.

## 1 Introduction

Over the last decade, unsolicited bulk e-mails, i.e., spams, have been dramatically increasing and they have been definitely recognized as a serious Internet threat. Many researches report that more than 90% of all Internet e-mails today are considered as spam[1], and they are abused for various purposes. Most spams assume the form of advertising or promotional materials, for example, those related to money, debt reduction plans, getting-rich-quick schemes, gambling opportunities, porn, health and so on[2]. Spams lead to many social problems in terms of productivity and IT infrastructure, and as reported in [3,4], the worldwide cost of spams was estimated to be well over US$10 billion in 2005.

On the other hand, recent spams mostly caused by various malwares (e.g., bots, worms) often contain URLs which navigate spam receivers to malicious Web servers for the purpose of malware infection. In addition, many malwares such as bots operate in cooperation with each other, and there are close links between malwares and malicious Web servers. For example, a botnet consists of many distributed bots connected worldwidely and it sends tremendous amount

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 556–564, 2009.

of spams to target e-mail users in large quantities by means of a command from the C&C server remotely controlled by their herder. Further, botnet herders possess their own malicious Web servers, from which they are able to propagate their well-crafted malwares to victims. There are a wide range of ongoing spam sending systems (e.g., unauthorized relay, dedicated MTA, Web mail, and bots) and diverse purposes of spam (e.g., malware infection, phishing, and e-mail address harvesting). Therefore, in order to cope with these complicated and unclear situations, it is strongly expected to investigate spams' sources and their destinations, and the correlation between them all together in an integrated manner.

There are many anti-spam strategies; however, their main goals are limited only to spam detection, or they take into account spams' sources and URLs' destinations separately[1,5,6]. In this paper, we propose a practical methodology for analyzing the overall flow of spam-based attacks in order to investigate the active relationship among spams, malwares and malicious Web servers. To this end, we construct two analyzers, i.e., source analyzer and destination analyzer, which analyze the characteristics of spams' sources and their eventual destinations, respectively. For the correlation analysis between them, we generate spams' profiles which contain the analysis results obtained by two analyzers. Furthermore, we have evaluated our method using double bounce e-mails obtained from our own SMTP server. The experimental results show that spams' sources and malicious Web servers are closely connected to each other, and the proposed method is highly effective to analyze the correlation between spams' sources and their eventual destinations.

The rest of the paper is organized as follows. In section 2, we give a brief description of the existing researches related to our research. In section 3, we present our approach, and the experimental results including their analysis are given in section 4. Finally, we present our concluding remarks.

## 2 Related Work

In [1], Anderson et al. have focused on the scam infrastructure that is nourished by spam and have described an opportunistic measurement technique called spamscatter that mines e-mails in real-time, follows the embedded link structure, and automatically clusters the destination websites using image shingling to capture graphical similarity between rendered sites.

In [5], Kreibich et al. have presented an inside look at how campaign orchestration takes place by analyzing the raw material used to produce spam, including textual templates employed for generating highly diverse spam instances. Their analysis shows that today's spamming business operates at a frightening scale without requiring sophisticated mechanisms to counter the anti-spam industry's efforts.

In [6], Kawakoya et al. have analyzed the characteristics of Web-based passive attacks which are driven by the URLs included in spams. They have developed a client-type honeypot to do this. During 15 days observation for a bulk of spam

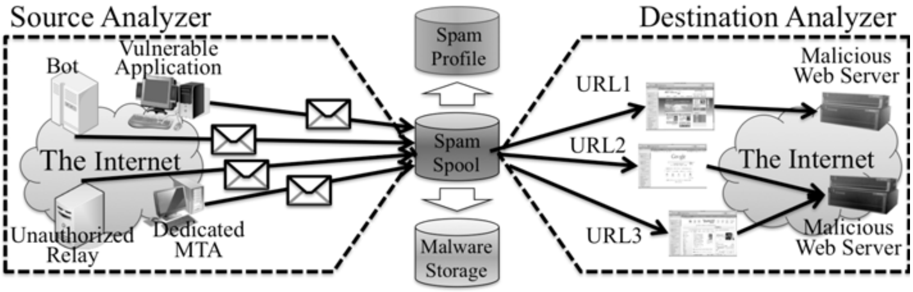


**Fig. 1.** Overall structure of the proposed method

sent to a mail server of Kyoto University, they found 409 malicious Web sites and 31,618 spam mail messages that had links to malicious websites.

Although these approaches are most closely related to ours, there is a fundamental limitation of their approach is that they have focused on the analysis of either spams' sources or their eventual destinations, so that it is quite difficult to analyze not only the overall flow of spams, but also the correlation between them.

## 3 Proposed Method

Figure 1 shows the overall structure of the proposed method, which consists of two analyzers (i.e., source analyzer and destination analyzer) and three repositories (i.e., spam spool, spam profile, and malware storage). In our method, we first collect spams from the Internet, and they are stored at a spam spool repository. In order to collect spams, we used double bounce e-mails described in Section 3.1. The source analyzer aims to identify the types of spam sending system and spam herder, and its analysis results are stored at a spam profile repository. The destination analyzer investigates which e-mail has URLs linked to malicious Web pages, and downloads their HTML contents and malwares if possible. Its analysis results are stored at two repositories, i.e., spam profile and malware storage. See Sections 3.2 and 3.3 for more detail.

### 3.1 Double Bounce E-Mail

During the previous years, many machine learning methods have been proposed for spam detection. However, these methods cannot accurately distinguish spams from normal e-mails. Further, the effective lifetime of the existing techniques is highly short, because spammers frequently change their modus operandi to compromise spam detection techniques. In other words, in order to maintain the effectiveness of spam filters, constant upgrades and new developments are essentially required. Considering the above, we do not capture our analysis data, i.e., spams, using the previous spam detection methods.

Double bounce e-mails indicate that they have no valid recipient address and return-path address. In the case of a normal e-mail, it contains one return-path address at least in its header field, even if a sender mistyped the recipient address to his/her e-mail. In this context, double bounce e-mails can be regarded as pure spam. This double bounce e-mail is quite similar to the concept of darknet which is an area of a routed, and allocated IP space where no active services or servers reside, and thus the incoming packets to it can be treated as abnormal ones. Darknet has been used for analyzing incidents in many researches[7], because many abnormal activities can be easily observed in the darknet. Similarly, we collect double bounce e-mails from the Internet and use them as our analysis data.

### 3.2 Source Analyzer

The source analyzer identifies the types of spam sending system and its herder. In our method, we classify the types of spam sending system into six categories: unauthorized relay, bot, Web mail, dedicated MTA, vulnerable application, and unidentified source. The identification mechanism is as follows.

- unauthorized relay: if an e-mail arrived at an internal SMTP server from more than two organizations, i.e., two different domain names, it is classified as unauthorized relay. Since in the case of large organizations, e.g., ISP, which often relay e-mails on their internal networks, we do not classify them in this category.
- vulnerable application: if the Received field of the e-mail header contains “localhost” or “127.0.0.1”, and the X-Mailer field represents an e-mail application such as IPB PHP Mailer, operating on the free bulletin board, then the corresponding e-mail falls in this category.
- bot: if an e-mail satisfies the following two conditions, then it is classified in this category. (1) There is no regular SMTP server in the Received field of the e-mail header, and (2) the Return-path field of the e-mail header is empty. If not, the hostname of the Return-path field is unmatched to the IP address which is obtained by reverse lookup of the MX record. Because bots try to directly connect to a target SMTP server, and it tends to inform a target SMTP server that their hostname is unmatched to the IP address.
- dedicated MTA: Dedicated MTA makes a regular form of the e-mail headers. If there are no suspicious fields in the e-mail header, we classify it as dedicated MTA.
- Web mail: if the Received field contains an IP address of Web mail servers, e.g., Hotmail, Yahoo, etc., then the corresponding e-mail is classified as a Web mail.
- unidentified source: if an e-mail does not belong to the above five types, and the Received field contains a suspicious value, e.g., a host connected to several different SMTP servers, then we classify the e-mail as an unidentified source.

The other purpose of the source analyzer is to classify the given spams into several groups based on similarity of the e-mail header. If the headers of two e-mails have the same pattern, i.e., the order of the header fields, they become the members of the same group. Considering the situation where RFC 2821[8] does not contain the comment about the order of the header fields, if the patterns of two e-mail headers are the same, the patterns can be regarded as being sent by the same SMTP engine. This also means that they originate from the same herder because in many cases SMTP engines basically have different operations.

For each e-mail, an entry including the IP addresses of its sender and receiver, an IP address of a relay SMTP server, a type of spam sending system and a type of spam herder, is inserted to the spam profile repository.

### 3.3 Destination Analyzer

As shown in Figure 2, the destination analyzer consists of four main parts : URL extractor, database, crawler, and evaluator. The Destination Analyzer first extracts the URLs from each e-mail, and then removes duplicate URLs using the URL extractor. We input each unique URL into the database, and then the crawler reads each of them one by one(①). The crawler attempts to connect to a website directly linked to the corresponding URL, and downloads available data(e.g., HTML contents, malwares) and new URLs, until there are no more URLs to be analyzed(②). The crawling results of ② are inserted into the database(③). The evaluator reads the data to be evaluated from the database, and evaluates whether the data are malicious or not(④). For this investigation, we use a dedicated software, i.e., SPIKE[9] which has three analysis modules:

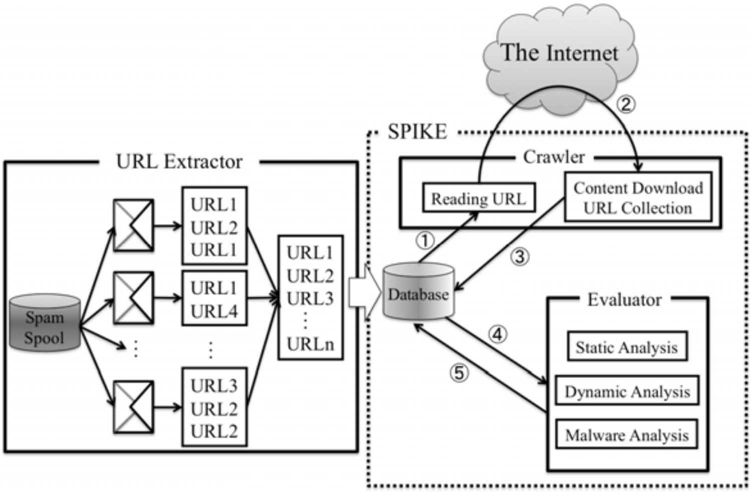


**Fig. 2.** Process of destination analyzer

static analysis module, dynamic analysis module and malware analysis module. The evaluation results of ④ are stored in the database(⑤).

We add these analysis results, e.g., whether an e-mail contains an URL linked to a malicious Web site or not, and IP addresses related to the URLs to each entry created in the source analyzer. Further, downloaded malwares are stored in the malware storage repository.

## 4 Experimental Results

### 4.1 Data Captured for Our Analysis

The experimental data we used were double bounce e-mails that arrived at our own SMTP server. We gathered about four days(April 16, 2009, to April 20, 2009) of double bounce e-mails, and captured 108,839 e-mails. From the original 108,839 e-mails, we filtered out some irregular e-mails from our evaluation data, and consequently we obtained 98,890 regular e-mails. We extracted 159,215 unique URLs from these regular e-mails.

### 4.2 Results and Analysis

We first evaluated 159,215 unique URLs using the destination analyzer from July 16, 2009, to July 29, 2009, and observed that 16 URLs were directly linked to a malicious Web page; however, there were no executable malwares. Further, among the regular 98,890 e-mails, 137 e-mails contained one of those 16 URLs. We divided 137 e-mails into 16 groups according to each URL. We denoted each group as $\{g_0, g_1, \cdots, g_{15}\}$. With respect to these 16 groups, we analyzed them by using the source analyzer. Table 1 shows the analysis results. From Table 1, we can observe that the 16 groups were sent by 6 different spam herders(i.e., a, b, c, d, e, and f), and 3 different spam sending systems(i.e., bot, vulnerable application, and unauthorized relay). We can predict that 6 different types of bots belonged to a single botnet, and they were under a controller (i.e., type

**Table 1.** Classification result by source analyzer

| | | Type of spam sending system | | | | | |
|---|---|---|---|---|---|---|---|
| | | bot | vulnerable application | unauthorized relay | dedicated MTA | Web mail | unidentified source |
| Type of spam herder | a | $g_0$, $g_1$, $g_2$, $g_3$, $g_6$, $g_{12}$ | - | - | - | - | - |
| | b | - | $g_5$, $g_{10}$, $g_{13}$ | - | - | - | - |
| | c | - | $g_8$, $g_9$, $g_{15}$ | - | - | - | - |
| | d | - | $g_7$ | - | - | - | - |
| | e | - | - | $g_{11}$ | - | - | - |
| | f | - | - | - | - | - | $g_4$, $g_{14}$ |

**Table 2.** Correlation between spams' sources and URLs' destinations

| Group | Sender IP | URL IP | Group | Sender IP | URL IP |
|---|---|---|---|---|---|
| $g_0$ | 62.h1.h2.206(RU) | 216.x1.x2.102(US) | $g_5$ | 78.n1.n2.165(TR) | 200.v1.v2.78(BR) |
| $g_3$ | 216.i1.i2.102(US) | 216.x1.x2.102(US) | $g_7$ | 89.o1.o2.97(RO) | 200.v1.v2.78(BR) |
| $g_1$ | 189.j1.j2.18(BR) | 68.y1.y2.143(US) | $g_8$ | 58.p1.p2.204(PK) | 200.v1.v2.78(BR) |
| $g_2$ | 78.k1.k2.8(RU) | 68.y1.y2.143(US) | $g_9$ | 95.q1.q2.158(RO) | 200.v1.v2.78(BR) |
| $g_6$ | 201.l1.l2.208(BR) | 68.y1.y2.143(US) | $g_{10}$ | 200.r1.r2.150(BR) | 200.v1.v2.78(BR) |
| $g_{12}$ | 80.m1.m2.94(IR) | 68.y1.y2.143(US) | $g_{13}$ | 78.s1.s2.76(LT) | 200.v1.v2.78(BR) |
| - | - | - | $g_{15}$ | 89.t1.t2.41(RO) | 200.v1.v2.78(BR) |

"a" of spam herder); further, 3 different types of e-mail sending applications operating on a free bulletin board were abused to send spams.

In order to verify this, we investigated the correlation between spams' sources and URLs' destinations as shown in Table 2. It should be noted that the parenthesis indicates the country code of IP address. From Table 2, we can see that 6 different types of bots are located in 4 different countries (i.e., RU, US, BR and IR), and the spams sent by them induced e-mail receivers to two malicious Web servers in the US. Furthermore, in our investigation, we found that 6 URLs shared a common string, i.e., "http://xxxx.com/resp/xxxx," in which only "xxxx" changes to a different string in each URL.

On the other hand, in the case of 7 groups (i.e., $g_5$, $g_7$, $g_8$, $g_9$, $g_{10}$, $g_{13}$, and $g_{15}$), we can observe that 7 machines located in 5 different countries (i.e., TR, RO, PK, BR, and LT) are compromised by the vulnerability of e-mail sending application. In fact, we found that 3 different applications, IPB PHP Mailer, vBulletin Mail via PHP and MyBB, were used for sending spams. Further, it is easy to see that 7 URLs included in spams are directly linked to a single malicious Web server located in Brazil. In addition, we observed that they have a quite similar URL pattern, i.e., "http://yyyy.fantasticzoneonline.at/," in which only "yyyy" changes to a different string in each URL. As a result, this means that these spams were sent by only a single herder, and 7 machines may be under his/her control.

In the case of two groups(i.e., $g_4$ and $g_{14}$), we classified them into the type of unidentified source because we observed that they tried to connect to two different SMTP servers(including our own SMTP server) from a single host, which is unnatural in a normal case. Finally, it is easy to see that e-mails sent by dedicated MTA were not observed in our experimental data. In general, dedicated MTA sends a large amount of spam to many SMTP servers in a short time, when administrators of dedicated MTA receive a request from their client. As such, their activities can be easily identified, and consequently their domain name is blacklisted. Since a dedicated MTA changes the corresponding domain name to another one so as to evade being blacklisted, the lifetime of a dedicated MTA is considered to be very short. This means that we should analyze spam as soon as possible, so that we can identify which e-mails are sent by the dedicated MTA.

**Table 3.** Evaluation result according to links-depth by destination analyzer

| | Links-depth | | |
|---|---|---|---|
| | 2 | 3 | 4 |
| # of input URLs | 4,629 | 4,629 | 4,629 |
| # of accessible URLs | 4,701 | 5,843 | 325,596 |
| # of malicious URLs | 9(0) | 6(0) | 1,181(12) |

The above results were obtained under the condition where the depth of links to follow Web pages is only 1. In other words, for each given URL, SPIKE connects to only the corresponding Web page, and never jumps to another Web page linked on that Web page. However, in many cases, Web-based attacks tend to be carried out through several steps, i.e., several Web pages. Because attackers have to lure victims, who are browsing on a famous website such as Google or Yahoo, to their malicious website, in which well-crafted malwares are embedded. Thus, we reevaluated 4,629 URLs that were accessible to connect, and the experimental results are shown in Table 3. From Table 3, we can understand that both the number of accessible URLs and the number of malicious URLs are rapidly increasing when the links-depth is 4. It is to be noted that the parenthesis indicates the number of URLs where executable malware was downloadable. These results signify that it is highly important to trace the overall flow of the Web-based attacks, and our approach provides the basic and expansive methodology which enables one to do that.

## 5 Conclusion and Future Work

In this paper, we have proposed a methodology for analyzing the overall flow of spam-based attacks in order to investigate useful correlations among spams, malwares and malicious Web servers. To this end, we have constructed spams' profiles, which enable us to identify not only the overall flow of spams but also the characteristics of spam sending system and spam herder and to identify which e-mail is connected to a malicious Web page; further, we have successfully provided the correlation between spams' sources and their eventual destinations.

We have evaluated the proposed methodology using double bounce e-mails obtained from our own SMTP server and have showed its effectiveness to analyze the correlation between spams' sources and their eventual destinations. Our key findings can be summarized as follows: (1) 16 URLs among 159,215 unique URLs extracted from 98,890 regular e-mails were directly linked to a malicious Web page, (2) 6 different types of bots belonging to a single botnet were located in 4 different countries, and spams sent by them lured e-mail recipients to only two malicious Web servers in the US whose URLs were quite similar. (3) 7 machines located in 5 different countries sent spams by abusing the vulnerability of an e-mail sending application operating on the free bulletin board, and spams' URLs were directly linked to a single malicious Web server located in Brazil, and (4) the number of accessible URLs and the number of malicious URLs were rapidly increasing when the links-depth was 4.

For the future work, we need to analyze further the correlation between URLs, i.e., Web pages and their sources with respect to the links-depth larger than 1. In addition, we will evaluate our methodology under a real-time operational environment. In order to analyze spam-based attacks more effectively, we have a plan to visualize their overall flow utilizing the IP address information of the spam profile repository, and perform correlation analysis between malwares stored at the malware storage repository and other type of malwares, e.g., malwares gathered by nicter(Network Incident analysis Center for Tactical Emergency Response)[7].

## References

1. Anderson, D.S., Fleizach, C., Savage, S., Voelker, G.M.: Spamscatter: characterizing Internet scam hosting infrastructure. In: Proceedings of the USENIX Security Symposium, Boston (2007)
2. Li, F., Hsieh, M.: An empirical study of clustering behavior of spammers and group based anti-spam strategies. In: Conference on Email and Anti-Spam 2006 (CEAS 2006), pp. 21–28 (2006)
3. Jennings, R.: The global economic impact of spam, 2005 report. Technical report, Ferris Research (2005)
4. Spira, J.: Spam e-mail and its impact on it spending and productivity. Technical report, Basex Inc. (2003)
5. Kreibich, C., Kanich, C., Levchenko, K., Enright, B., Voelker, G., Paxson, V., Savage, S.: Spamcraft: an inside look at spam campaign orchestration. In: Proceedings of the 2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET 2009), Boston (2009)
6. Kawakoya, Y., Akiyama, M., Aoki, K., Itoh, M., Takakura, H.: Investigation of spam mail driven Web-based passive attack. IEICE Technical Report, ICSS2009-5, May 2008, 21–26 (2009)
7. Nakao, K., Inoue, D., Eto, M., Yoshioka, K.: Practical correlation analysis between scan and malware profiles against zero-day attacks based on darknet monitoring. IEICE Transactions on Information and Systems E92D(5), 787–798 (2009)
8. RFC 2821, http://www.ietf.org/rfc/rfc2821.txt
9. Hosihzawa, Y., Kawamorita, K., Tachikawa, T., Kamizono, M.: A Proposal for autonomous crawling client honeypot. IEICE Technical Report, IA2009-3, ICSS2009-11, June 2009, 13–17 (2009)

# A Proposal of Malware Distinction Method Based on Scan Patterns Using Spectrum Analysis

Masashi Eto[1], Kotaro Sonoda[1], Daisuke Inoue[1], Katsunari Yoshioka[2], and Koji Nakao[1]

[1] National Institute of Information and Communications Technology
4-2-1, Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan
[2] Yokohama National University
79-7 Tokiwadai, Hodogaya-ku, Yokohama 240-8501, Japan

**Abstract.** Network monitoring systems that detect and analyze malicious activities as well as counter them, are becoming increasingly important. As malwares, such as worms, viruses, and bots, can inflict significant damages on both the infrastructure and the end user, technologies for identifying such propagating malwares are in great demand. In the large-scale darknet monitoring operation, we can see that malwares have various kinds of scan patterns that involves choosing destination IP addresses. With a focus on such scan patterns, this paper proposes a novel concept of malware feature extraction and a distinct analysis method named "*SPectrum Analysis for Distinction and Extraction of malware features* (*SPADE*)."Through several evaluations using real scan traffic, we show that SPADE has the significant advantage of recognizing the similarities and dissimilarities between the same and different types of malwares.

## 1 Introduction

Malwares are spread all over the Internet and they often lead to serious security incidents that can cause significant damage to both the infrastructure and end users. As countermeasures, a number of ongoing network monitoring projects are already in their operational phase. Many of these projects are concentrating on the event analysis that provides statistical data, such as rapid increase in access on certain port numbers, by using network event monitoring. Particularly, it is becoming popular to monitor a dark address space (darknet), which is a set of globally announced unused IP addresses [1,2].

In order to identify the root causes of network events observed on darknets, we have started the *Network Incident analysis Center for Tactical Emergency Response (nicter)* project, with the goal of achieving an integrated analysis of security incidents on large networks [3,4]. Our present focus is particularly on detecting and identifying the propagation of malwares such as worms, viruses, and bots, which can infect remote hosts through fundamental propagation steps,

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 565–572, 2009.

such as *scan* → *exploit code* → *malware download*, by exploiting the vulnerabilities of operating systems or server applications of the targeted hosts. One of the purposes of nicter is to reveal certain malware species, which infect attacking hosts, only through their scan patterns. As a first step to realize this purpose, this paper aims at providing a method to distinguish attacking hosts observed on darknets.

Through our large-scale darknet monitoring operation, we have learned that malware uses various kinds of scan patterns, that involves choosing destination IP addresses [5,6,7,8], such as regular increment or random determination of destination IP addresses for each packet. Since these scan patterns resemble a signal waveform, we applied the discrete Fourier Transform (DFT) algorithm to the feature extraction and distinction method.

In this paper, we propose a malware feature extraction and distinction method called *SPectrum Analysis for Distinction and Extraction of malware features* (*SPADE*), which analyzes the scan traffic data from a pair of attacking hosts and derives the correlation coefficient between them.

The rest of this paper is organized as follows: Section 2 introduces the background of this research and related works. The SPADE algorithm is explained in Section 3. Several evaluations with actual darknet traffic are presented in Section 4. Finally, Section 5 presents the conclusions and future work.

## 2 Background

As a background of this research, we first explain the darknet monitoring and its advantage. Second, we show that malwares have various scan patterns according to our long-term and large-scale darknet monitoring, which can be used to classify the malwares. Third, we mention some related works.

### 2.1 Darknet Monitoring

A darknet is a set of globally announced unused IP addresses and using it is a good way of monitoring network attacks, such as malwares' scans. A big advantage of darknet monitoring is that there is no legitimate host using these addresses; we can thus consider all incoming traffic to be a consequence of some kind of malicious activity or the result of misconfigurations. As the principal darknet monitoring method, we deployed *black hole sensors*, which quietly monitor incoming packets without ever responding to the opposite hosts. By using these black hole sensors in the darknet IP domains, we could observe emerging network attacks, including malware-initiated network scan, malware infection behavior, and DDoS backscatters.

### 2.2 Scanning Patterns of Malwares

According to our long-term and large-scale darknet monitoring and some previous researches [5,6,7,8], malwares use various kinds of scan patterns. These patterns

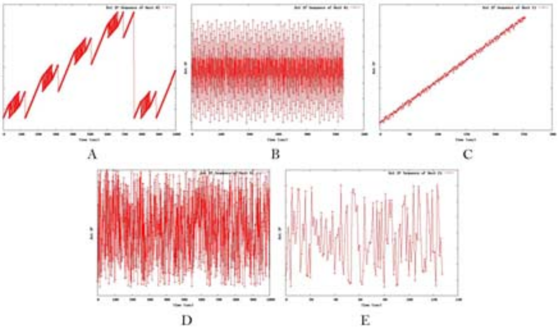


**Fig. 1.** Examples of Malware Scanning Patterns

can be important clues to classify malwares. Figure 1 shows the examples of typical scan patterns that are observed by nicter's black hole sensors. These graphs represent scan patterns of five individual attacking hosts $A$-$E$ that are probably infected by different types of malwares. In each graph, the $X$ axis represents the packet arrival sequence and the $Y$ axis represents the value of the destination IP address of our black hole sensor that monitors a network with /16 subnet mask.

The examples state that there are various scan patterns, such as regular increment pattern ($A$, $C$), random determination pattern ($D$, $E$), and specific rule-based pattern that appears to be a random pattern ($B$). Through these observations, we found that the scan patterns of malwares have their own individual characteristics, and also found that many of the patterns appear to have the natural periodicity, as if they are signal waveforms. Consequently, we apply a spectrum analysis method to the scan traffic in order to extract their characteristics and to classify them.

### 2.3 Related Works

In the context of spectrum analysis, Mitra et al. [9] proposed an anomaly detection method based on spectrum analysis. They focused on the harmonic structure of the traffic data spectrum obtained by Fourier transform and wavelets so as to detect DDoS and bottleneck traffic. Meanwhile, Yu et al. [10] applied a spectrum analysis method for detecting slow scan worms. The objective of these studies was to detect the presence of anomalous activities in normal traffic on a live network by means of analyzing the fluctuation of traffic volume. In contrast, since our basic tactics involves black hole monitoring, where most of the observed activities are assumed to be anomalous, we aim at establishing a fine-grained distinction method of attacking hosts based on their scan patterns rather than merely determining whether the activities are malicious or not.

## 3 SPADE: Malware Correlation Method Based on Spectrum Analysis

In this section, the SPADE algorithm is proposed, which is a malware feature extraction and correlation method based on spectrum analysis. SPADE begins with the application of discrete Fourier transform (DFT) to a series of destination IP addresses and ends at the derivation of a correlation coefficient between two different series of data.

### 3.1 Algorithm of SPADE

The main processes of SPADE algorithm are as follows.

1. Hamming Window Function for Series Data
   As a preparation of subsequent processes, SPADE applies the *Hamming window function* to the original series data of destination IP addresses in order to emphasize the characteristics of the oscillations. Specifically, where the original series data $X$ has $N$ elements, the following formula is applied to each element $X(n)$;

   $$X'(n) = \{0.54 - 0.46\cos(\frac{2\pi n}{N})\}X(n) \tag{1}$$

   It should be noted that the parameters (0.54 and 0.46) were defined as the constant number of Hamming window function [11].
2. Discrete Fourier Transform
   SPADE applies DFT to the series data and derives a spectrum, which represents the characteristics of the original scanning behavior.
3. Removal of High-Frequency Bands
   Minor phenomena such as packet losses and disorders in the packet arrival sequence appear in high-frequency bands. Therefore, in order to avoid these influences, which are the result of the degradation of network conditions, SPADE removes frequency bands higher than the vertical threshold A.
4. Extraction of Maximum Value Indices
   In this step, SPADE extracts the peaks of frequency components, namely, those whose magnitudes are higher than neighboring components, since high-level components in a spectrum are dominant in characterizing the original scanning behavior. Furthermore, with this process, SPADE reduces the number of samples for the subsequent calculations.

   It is to be noted that in the following steps, SPADE treats a collective set of indexes I (and not power levels) of the components that are selected in this step.
5. Removal of Fundamental Frequency
   The purpose of this step is to adjust the number of cycles of oscillations among different attacking hosts. Even if some attacking hosts have the same scanning behavior, the packet counts from them may differ widely because of the differences in the conditions of each network sensor. To resolve this, SPADE removes the fundamental frequency from the spectrum while maintaining its harmonic structure. The normalized index values $(N_i)$ are derived by dividing each index value $(I_i)$ by the index value of the fundamental frequency $(I_p)$ that has the highest power level in the spectrum, as shown by formula (2).

$$N_i = \frac{I_i}{I_p} \tag{2}$$

6. Standardization of Harmonic Structure
   In this step, SPADE standardizes the weight of each value in the normalized index values $N$ and derives a series of deviation values $S$. As mentioned in the previous step, the number of observed packets is dependent on the environmental conditions of the sensor. To neutralize the differences in environmental conditions, SPADE computes a series of deviation values ($S$) of $N$. For a series of index values $N$ that has $n$ samples, the standard deviation $SD_N$ is derived by formula (3).

   $$SD_N = \sqrt{\sum_{i=1}^{n} \frac{(N_i - M)^2}{n}} \tag{3}$$

   Here, $M$ is the average of the all index values. The deviation value of each index value ($S_i$) is computed from the standard deviation ($SD_N$) by formula (4).

   $$S_i = \frac{N_i - M}{SD_N} \tag{4}$$

7. Synchronization and Alignment of Two Series of Data
   As the last step of the preparation process, we must synchronize two independent series of data and adjust their lengths. In this step, SPADE aligns two series based on the index values of their base frequencies. The missing space resulting from the differences in the series lengths is padded with zeros. Thus, the two series of data are synchronized and aligned, ready to be compared with each other.
8. Derivation of Correlation Coefficient
   Finally, SPADE computes the correlation coefficient between two independent series of data. The correlation coefficient $C_{\alpha\beta}$ between series $S_\alpha$ and $S_\beta$ is derived by formula (5).

   $$C_{\alpha\beta} = \frac{1}{n} \sum_{i=1}^{n} S_{\alpha i} \times S_{\beta i} \tag{5}$$

   The correlation coefficient ranges from -1 to 1. An absolute value of coefficient approaches 1 with increasing similarity between the original scan data, and approaches 0 with decreasing similarity.

A sample result of the SPADE algorithm is shown in Figure 2, where two attacking hosts $A_1$ and $A_4$ are analyzed. As shown in Figure 2-(1) and (2), these hosts have similar scan patterns that oscillate destination IP addresses based on a specific rule although it appears to be determined randomly. Figure 2-(3) shows that the picked up maximum values (indicated by × and +) in each spectrum overlap on six points. Figure 2-(4) shows that the correlation coefficient between these hosts is 0.80.

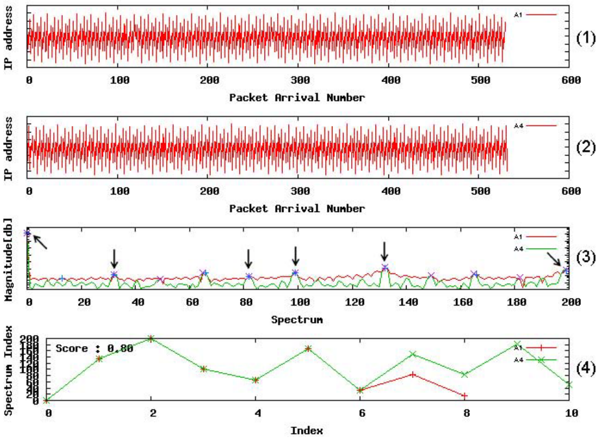


**Fig. 2.** Result of Correlation between $A_1$ and $A_4$

# 4 Evaluation

The purpose of SPADE is to extract the features of scan patterns from individual attacking hosts and derive the similarities or dissimilarities among them. Consequently, this section provides some evaluations using practical traffic data observed by nicter's black hole sensors in December 2008. High correlation coefficients are expected among hosts infected by the same type of bot, while low correlation coefficients are derived among hosts infected by different types of bots.

## 4.1 Analyses among Same Bots

At first, SPADE analyzed five attacking hosts ($A_1$ - $A_5$) in the same botnet, which sophisticatedly fluctuated destination IP addresses, and derived the correlation coefficients in all combinations (Table 1). There were six combinations whose coefficients were over 0.70, among the ten combinations excluding self-comparison, and the coefficient average of those ten combinations was 0.73.

**Table 1.** Correlation Results among Hosts in the Same Botnet (1)

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ |
|---|---|---|---|---|---|
| $A_1$ | 1.00 | 0.74 | 0.61 | 0.80 | 0.59 |
| $A_2$ | – | 1.00 | 0.71 | 0.92 | 0.60 |
| $A_3$ | – | – | 1.00 | 0.77 | 0.94 |
| $A_4$ | – | – | – | 1.00 | 0.65 |
| $A_5$ | – | – | – | – | 1.00 |

As a result, we found that most of the correlation coefficients among hosts in the same botnet were higher than at least 0.70. This signifies that SPADE can recognize hosts infected by the same type of bot.

### 4.2 Analyses among Different Bots

Secondly, five hosts were analyzed, which belonged to different botnets and had significantly dissimilar scan patterns. As shown in Table 2, there was only one combination whose correlation coefficient was over 0.70 among the ten combinations excluding self-comparison, and the coefficient average of those ten combinations was 0.41. According to this result, we found that SPADE can distinguish hosts that were infected by different types of bots.

**Table 2.** Correlation Results among Hosts in Different Botnets

| | $A_1$ | $B_1$ | $C_1$ | $D_1$ | $E_1$ |
|---|---|---|---|---|---|
| $A_1$ | 1.00 | 0.31 | 0.33 | 0.87 | 0.59 |
| $B_1$ | – | 1.00 | 0.28 | 0.31 | -0.01 |
| $C_1$ | – | – | 1.00 | 0.46 | 0.41 |
| $D_1$ | – | – | – | 1.00 | 0.57 |
| $E_1$ | – | – | – | – | 1.00 |

### 4.3 Consideration

The evaluation results showed that SPADE derived higher correlation coefficients between each combination within the same botnets, while lower correlation coefficients were derived within different botnets. This signifies that SPADE almost successfully extracted malwares' features and distinguished them. However, in some cases, the results also showed that the correlation coefficients widely varied from 0.59 to 0.98 even though the hosts belonged to the same botnet. Therefore, we need to improve the algorithm in order to extract malwares' feature more efficiently.

## 5 Conclusion and Future Work

In this paper, by focusing on the oscillations of the destination IP addresses of scan packets, we proposed the concept of malware feature extraction, and implemented and evaluated a distinct analysis method (SPADE) that applied a spectrum analysis methodology. Our contribution is to realize a fundamental technology to grasp the general trend of malwares propagation only from their scan traffic data. In other words, we proposed SPADE algorithm that employed the discrete Fourier transform (DFT). Through several evaluations, we showed that SPADE almost successfully extracted malwares' features and distinguished them.

Although we applied the destination IP addresses of scan packets in our algorithm, we have to further consider other parameterized characteristics such as the source/destination port numbers and the interval time of packet arrival. As a future work, we shall attempt to establish more multifaceted analysis techniques covering such parameters based on the SPADE algorithm.

## References

1. Bailey, M., Cooke, E., Jahanian, F., Nazario, J., Watson, D.: The Internet Motion Sensor: A Distributed Blackhole Monitoring System. In: The 12th Annual Network and Distributed System Security Symposium, NDSS 2005 (2005)
2. Moore, D.: Network Telescopes: Tracking Denial-of-Service Attacks and Internet Worms around the Globe. In: 17th Large Installation Systems Administration Conference, LISA 2003 (2003)
3. Inoue, D., Eto, M., Yoshioka, K., Baba, S., Suzuki, K., Nakazato, J., Ohtaka, K., Nakao, K.: nicter: An Incident Analysis System Toward Binding Network Monitoring with Malware Analysis. In: WOMBAT Workshop on Information Security Threats Data Collection and Sharing, pp. 58–66 (2008)
4. Nakao, K., Yoshioka, K., Inoue, D., Eto, M.: A Novel Concept of Network Incident Analysis based on Multi-layer Observations of Malware Activities. In: The 2nd Joint Workshop on Information Security (JWIS 2007), pp. 267–279 (2007)
5. Filiol, E.: Malware Pattern Scanning Schemes Secure Against Black-box Analysis. Journal in Computer Virology 2, 35–50 (2006)
6. Zou, C., Gong, W., Towsley, D.: Code red worm propagation modeling and analysis. In: 9th ACM conference on Computer and communications security, pp. 138–147 (2002)
7. Chen, Z., Gao, L., Kwiat, K.: Modeling the spread of active worms. In: Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2003, vol. 3. IEEE, Los Alamitos (2003)
8. Garetto, M., Gong, W., Towsley, D., di Elettronica, D.: Modeling malware spreading dynamics. In: Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2003, vol. 3. IEEE, Los Alamitos (2003)
9. Mitra, U., Ortega, A., Heidemann, J., Papadopoulos, C.: Detecting and Identifying Malware: A New Signal Processing Goal. Signal Processing Magazine 23, 107–111 (2006)
10. Yu, W., Wang, X., Calyam, P., Xuan, D., Zhao, W.: On Detecting Camouflaging Worm. In: 22nd Annual Computer Security Applications Conference (ACSAC 2006), pp. 235–244. IEEE Computer Society. Washington (2006)
11. Harris, F.: On the use of windows for harmonic analysis with the discrete Fourier transform. Proceedings of the IEEE 66, 51–83 (1978)

# A Transductive Neuro-Fuzzy Force Control: An Ethernet-Based Application to a Drilling Process

Agustin Gajate[1], Rodolfo Haber[1], and Pastora Vega[2]

[1] Instituto de Automatica Industrial, Spanish Council for Scientific Research
Carretera de Campo Real km. 0,200, 28500 Arganda del Rey, Spain
[2] Departamento de Informatica y Automatica, Universidad de Salamanca
Pza. de los Caidos s/n, 37008 Salamanca, Spain

**Abstract.** This paper presents the application of a neural fuzzy inference method to the field of control systems using the internal model control paradigm (IMC). Through a transductive reasoning system, a neuro-fuzzy inference system enables local models to be created for each input/output set in the system at issue. These local models are created for modeling the direct and inverse dynamics of the process. The models are then applied according to IMC paradigm. In order to demonstrate the benefits of this technique for control systems, it is applied for networked cutting force control in a high-performance drilling process. A comparative study between a well-established neuro-fuzzy technique and the suggested method is performed.

**Keywords:** Transductive modeling, Neuro-fuzzy inference, Internal model control, Networked control, High-performance drilling.

## 1 Introduction

Recent years have been characterized by the development of new paradigms in the field of Artificial Intelligence (AI). One of the research disciplines that has spawned the most developments is soft computing, where computing, reasoning and decision-making make the most out of tolerance for imprecision, uncertainty and approximate reasoning, in order to yield better solutions [1]. The hybridization of fuzzy logic with neural networks is the most well-established and best-known method within soft computing.

By the late nineties, several hybrid neuro-fuzzy systems had already been developed, which may be separated into two major groups: neural networks endowed with the ability to handle fuzzy information [fuzzy-neural networks (FNN)] [2], [3], and fuzzy systems combined with neural networks in order to enhance certain desirable characteristics [neural-fuzzy systems (NFS)] [4], [5]. A combination of evolutionary computation with a neuro-fuzzy system is proposed too in several works.

Most of the above-mentioned neuro-fuzzy strategies apply inductive reasoning systems. In inductive reasoning the key issues is to find a general model drawn from the entire set of input/output data representing the whole system.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 573–582, 2009.

In contrast there are transductive reasoning methods which generate a model at a single point of the workspace. For each new datum that has to be processed, the closest examples are selected among the known data, with the goal of creating a new local model that dynamically approximates the process in its new state as close as possible. The main issue is therefore how to assign more weight to the specific information related with the datum to be processed than to the general information provided by the entire training set [6].

Transductive methods have some advantages over inductive methods, because sometimes creating a valid model for the entire space or region of operation is a difficult task, yielding insufficient performance in some cases. In addition, these strategies are capable of functioning correctly with a small training set. Transductive reasoning methods have been applied to text recognition applications, time series prediction and medical diagnosis applications. The use of local models in control tasks first appeared in 1990 for controlling some movements in robotics [7], [8]. However, on the basis of reviewed literature, applications in the field of control manufacturing processes as well as to improve machining processes have not been previously reported.

On the other hand, the design of intelligent controllers necessarily requires control schemes and methods that guarantee the desired features. The internal model control (IMC) paradigm has been applied successfully in intelligent control systems. It is considered a consolidated method for designing fuzzy, neural and neural-fuzzy controllers for process control and in addition a method that guarantees a good dynamic response and reliable behavior in the presence of disturbances [9]. Thus, the characteristics discussed above make the internal model control paradigm and NFI suitable strategies [10], whose synergy can handle processes that display nonlinear and time-variant behavior and where classic techniques have not yielded the desired results.

Several works reported in the literature address the design and implementation of intelligent control systems for machining processes with the objective of optimizing these processes. Of all the machining operations, the drilling process is precisely one of the processes that has received less attention with a view to process improvement through the application of control techniques. This paper is focused on increasing the material removal rate as well as enhancing useful tool life through networked control of the cutting force by means of real-time modification of the tool's own feed rate.

## 2 Transductive Neuro-Fuzzy System

The transductive neuro-fuzzy inference system (NFI) involves the creation of unique local models for each subspace of the problem, using the Euclidean distance [11]. The system's inputs can be treated in different kinds of physics units but the normalization is recommended. In this paper each input data is normalized subtracting the mean and dividing by the standard deviation of the set of known data or training set. After the normalization, the personalized local model is then created using data from the training set that are the closest to each new input datum. The Euclidean distance is used for selecting this data subset. The size of this subset ($N_q$) is an input parameter for the algorithm.

NFI uses a Mamdani-type inference method and fuzzy membership functions are typically Gaussian-type. This type of membership function that is derivable enables the use of supervised learning algorithms such as back-propagation algorithms. The evolving clustering method (ECM) is used to create membership functions and fuzzy rules [11]. It consists in a single-iteration algorithm for the dynamic on-line clustering of a data set. For all following data, on the basis of Euclidean distances and the clustering threshold value ($D_{thr}$), the algorithm either adds each datum to an existing set (updating the center and the radius of the set) or creates a new set. For that purpose, the center of the set is taken as the center of the Gaussian function, and the radius is taken as the width.

Considering $P$ inputs, one output and $M$ fuzzy rules initially defined by the clustering algorithm, the $lth$ rule has the form:

$R_l$: If $x_l$ is $\phi_{l1}$ and $x_2$ is $\phi_{l2}$ and…$x_P$ is $\phi_{lP}$, then $y$ is $\gamma_l$. (Cluster $l$)

$$\phi_{lj} = \alpha_{lj} \exp\left[-\frac{(x_{ij}-m_{lj})^2}{2a_{lj}^2}\right] \tag{1}$$

$$\gamma_l = \exp\left[-\frac{(y-n_l)^2}{2{\delta_l}^2}\right] \tag{2}$$

where $m$ and $n$ are the centers of the Gaussian functions for the inputs and outputs, $a$ and $\delta$ are the widths, $i = 1,2,\dots,N_q$ is the index representing the number of closest neighbors, $j = 1,2,\dots,P$ represents the number of input variables, and $l = 1,2,\dots,M$ represents the number of fuzzy rules.

The centers $m$ and $n$ and the widths $a$ and $\delta$ are obtained as the result of the ECM algorithm, while the parameter $\alpha_{lj}$ is chosen by design ($\alpha_{lj}$=1) and represents the weight of each of the input membership functions. These parameters are adjusted with the back-propagation algorithm.

Using the center of area defuzzification method, the output of the NFI for an input vector $\overline{x}_i = [x_1, x_2, \dots, x_p]$ is calculated as follows:

$$O(\overline{x}_i) = \frac{\sum_{l=1}^{M} \frac{n_l}{\delta_l^2} \prod_{j=1}^{P} \alpha_{lj} \exp\left[-\frac{(x_{ij}-m_{lj})^2}{2a_{lj}^2}\right]}{\sum_{l=1}^{M} \frac{1}{\delta_l^2} \prod_{j=1}^{P} \alpha_{lj} \exp\left[-\frac{(x_{ij}-m_{lj})^2}{2a_{lj}^2}\right]} \tag{3}$$

The system uses input/output data of the closest training data $[\overline{x}_i, t_i]$ and the goal is to minimize the following target function:

$$E = \frac{1}{2} v_i \left[f(\overline{x}_i) - t_i\right]^2 \tag{4}$$

where $v_i$=1-$(d_i - \min(\overline{(}d)))$ with $i = 1,2,\dots,N_q$ and $\overline{(}d)$=$\left[d_1, d_2, \dots, d_{N_q}\right]$ as the vector of distances calculated in the first step. Fig.1 shows the block diagram of the implemented algorithm.

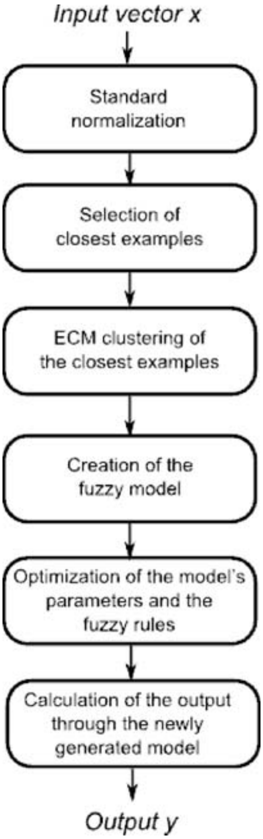

Input vector x
Standard normalization
Selection of closest examples
ECM clustering of the closest examples
Creation of the fuzzy model
Optimization of the model's parameters and the fuzzy rules
Calculation of the output through the newly generated model
Output y

**Fig. 1.** Block diagram of the NFI algorithm [11]

## 3 The High-Performance Drilling Process

Drilling is one of the most intensely used processes in the manufacture of aeronautic and automobile components in addition to the manufacture of moulds and dies. One of the main targets of manufacturing companies is to reduce production times and increase the quality of their products. Lower cycle times in a drilling process are directly related to increasing the material removal rate without damaging the cutting tool and without affecting the finish quality. In the high-performance drilling process the work is performed at very high cutting speeds at or near the limitations of the material, tool and machine tool.

Drilling force is the most important variable in the drilling process. Higher feed rates increase the material removal rate along with an increase in drilling force. However, higher cutting forces and drilling torques yield negative effects as well, such as rapid drill wear, tool vibration and the risk of catastrophic tool failure. Thus, it is important not only to maintain a constant drilling force, but also to obtain good closed-loop dynamic behavior without oscillations and overshoots, thus increasing useful tool life.

Traditional control methods still have much room for improvement and thus far have not yielded the expected improvement of the drilling process on the basis of linear models. If a linear model is available, designing a PID controller is a straightforward affair. However, sometimes technical constraints make it impossible to carry out experimental identification and modeling, or the validity of the linear model is very limited. The performance of the linear controller deteriorates as the linear model's behavior diverges from the actual process's behavior.

## 4 Internal Model Control

Internal model control (IMC) is an intensely used, well-established technique for designing intelligent controllers. This closed-loop control scheme explicitly uses a model of the dynamics of the plant to be controlled situated in parallel with the plant. Furthermore, it also contains another model of the inverse of the plant's dynamics situated in series with the process and acting as a controller.

One of the advantages of this control scheme is that dynamic analysis and robustness properties can be easily checked. However, inverting nonlinear models is not an easy task, and there may be no analytical solutions. Another associated problem is that inverting the process model can lead to unstable controllers when the system is a non-minimum phase system.

In this work, the inverse model $G'_M$ (5) and the direct model $G_M$ (6) are output error (OE) models. The NFI algorithm is used to create the models (both direct and inverse) on line. With each new input into the control scheme, both models are calculated. Using this neuro-fuzzy inference technique, the creation of the inverse model proves to be simpler and always offers a solution.

$$f(k) = G'_M(F(k), F(k-1), f(k-1)) \tag{5}$$

$$F(k) = G_M(f(k), f(k-1), F(k-1)) \tag{6}$$

where $F(k)$ is the cutting force estimated by the direct model and $f(k)$ is the feed rate calculated by the inverse model.

Once the models have been obtained, a low-pass filter is included in the control scheme. The filter is incorporated in the control system to reduce the high-frequency gain and to enhance the systems' robustness. It also works to soften fast, brusque signal changes, thus improving the controllers' response.

The direct model must be trained to learn the process's dynamics. A NFI system is used with a training set made up of input/output data, where the input is the feed rate, while the cutting force is used as the output variable (Fig.2a). In order to calculate the inverse model, instead of inverting the direct model found analytically, another NFI system is used, whose training set contains data with cutting-force values as the input and feed rate values as the output. This way,

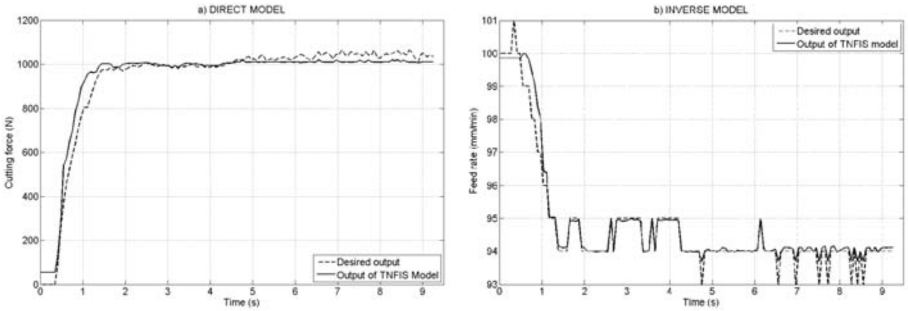


**Fig. 2.** a) Direct model and b) inverse model of the high-performance drilling process yielded by NFI

the system succeeds at learning the inverse dynamics of the high-performance drilling process (Fig.2b).

The training data for both the direct model and the inverse model were obtained from real drilling operations with test pieces made of A395 material (ASTM) under actual cutting conditions. Nevertheless, the training data set does not have to be very extensive, because representative values of each operating region are enough. The accuracy of the models depends on the choice of certain parameters of the NFI algorithm, such as the number of closest neighbors, the maximum number of iterations and the learning rate of the backpropagation algorithm, and the set-clustering threshold value (parameter of the clustering algorithm used). When choosing these parameters, the goal is to find a tradeoff between the accuracy and the quality of the dynamic response of the local models.

## 5 Experimental Results

In order to validate the networked intelligent control system by means of the suggested approach, it was decided to introduce the NFI-IMC in a real high-performance drilling process. Drilling processes are conducted in a machine tool Kondia HS1000 equipped with an open computer numerical control (CNC) Sinumerik 840D. A personal computer (PC1) is connected with the open CNC via a Profibus network. PC1 has a Windows 2000 operating system and performs three tasks. The first task is to measure force. The force is directly measured from PC1 with a Kistler 9257B dynamometer at a sampling frequency of 5kHz. The second task is communication with a second personal computer, PC2, via Ethernet with standard CORBA middleware. Filtered force $F$ and other parameters and variables (e.g., spindle speed and depth of drill) are passed to PC2 in this way. The third task is to receive the control signal computed by PC2 with the data interface and synchronization tasks performed by commercial software (Labview, NC-DDE application) over Ethernet.

User interface, data visualization and record, and the IMC neuro-fuzzy system are implemented in PC2 (Networked control was done through PC2). Free distribution software (RTAI-Linux and Qt) is used in this level of the developed platform. PC2 cannot be connected directly to the CNC due to proprietary CNC software constraints. NFI-IMC was developed in C++ and the control scheme was incorporated to PC2. The NFI-IMC system processed the force measurement, calculated the control signal ($f$) and sent it to PC1, which later modified the corresponding variable in the open-architecture CNC.

Network-induced delays are usually not known a priori and are difficult to measure and estimate on line. The maximum delay, including network-induced delays and process dead time, can be estimated thus:

$$[\tau_{SC} + \tau_{CA} + \tau_P]_{MAX} = 0.4s \tag{7}$$

where $\tau_{SC}$ is the delay from the dynamometric sensor to PC1, $\tau_{CA}$ is the delay from PC1 to the CNC and $\tau_P$ is the intrinsic dead time of the drilling process.

It is important to note that the delay due to switched Ethernet is not remarkable for many real-time applications, thus indicating a very promising alternative for networked control systems.

$$\tau_{S2} + \tau_{A2} = 0.005s \tag{8}$$

where $\tau_{S2}$ is the delay from PC1 to PC2, $\tau_{A2}$ is the delay from PC2 to PC1.

Different drilling tests were conducted on A564 precipitation-hardening stainless steel test pieces (martensitic steel). This is a material used heavily in the naval and aerospace industry. The optimum conditions recommended by tool manufacturers for drilling are: 780 rpm (spindle speed), 2000 N (reference value of force) and 93 mm/min (feed rate[1]). The experiments were carried out using a Sandvik R840-1000-30-A0A 10-mm-diameter tool of solid hard metal with a TiN/TiAlN coating.

The NFI algorithm parameters chosen are the same for the direct model and the inverse model (with the exception of the training sets, which represent different dynamics although they contain the same data). The chosen number of closest neighbors ($N_q$) is 5, and the number of iterations of the back-propagation algorithm is 20, with a learning rate of 0.001. The whole number of input and output membership functions is directly related with the number of closest neighbors ($N_q$), and therefore the maximum number of membership functions is 5 but changes dynamically according each new datum and the ECM algorithm. The threshold value selected for set generation in the ECM algorithm is $D_thr$=1. According with these parameters and models (5) and (6), the mean time elapsed between data input and control signal computation by NFI-IMC was 2.3ms on the basis of an Intel Core 2 CPU (CPU 6400, 2.13 GHz, 0.98 GB RAM) and Windows XP operating system.

A comparative study was performed with another very well-known neuro-fuzzy technique (ANFIS) [4] in an IMC scheme reported in [12]. The summary of

[1] Feed rate is set initially in the CNC operating program in order to begin part program execution, and later it is manipulated by the NFI-IMC control system.

**Table 1.** Neuro-fuzzy algorithms in internal model control paradigm

| Algorithm | ANFIS | NFI |
|---|---|---|
| System | Single-input/Single output | Single-input/Single output |
| Membership functions type | Gaussian | Gaussian |
| Number of membership functions | 2 | Variable each run (maximum 5) |
| Inference system | Takagi-Sugeno | Mamdani |
| Number of rules | 2 | Variable each run (maximum 5) |
| Iterations | 5 | 20 |
| Learning rate | 0.01 | 0.0001 |
| Training algorithms | Back propagation | Back propagation |
| Training data set | 139 samples | 139 samples |
| Validation data set | 134 samples | 5 samples |
| Training time | 134.4 ms | 2.06 ms |
| Mean computing time | 0.01 ms | 2.3 ms |

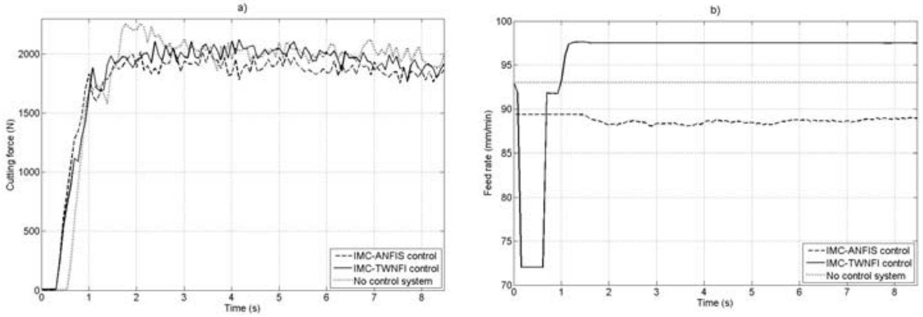


**Fig. 3.** a) Real system response and b) Control signal in operation with A564

the main characteristics of both neuro-fuzzy systems is shown in table 1. In order to compare both systems, the integral of time weighted absolute error ($ITAE$), the integral of the time square error ($ITSE$) and the integral of the square time multiplied by square error ($IT^2SE$) were used as performance indices. The overshoot ($Ovt$) was also included in the comparative study due to the influence of the transient dynamic on the useful tool life. The results of the experimental tests are shown in Fig.3 and in table 2. The dynamic behavior of ANFIS-IMC and NFI-IMC seems to be similar at first glance, but the NFI-IMC yields less overshoot than ANFIS-IMC. Furthermore, the NFI-IMC has better performance in $ITAE$, $ITSE$ and $IT^2SE$ criteria than the ANFIS-IMC strategy. In general, NFI-IMC strategy outperforms the ANFIS-IMC approach.

In addition, there is a significant difference at the beginning and at the end of the operation. This means that the entry of the drill on the workpiece is much smoother and that the evacuation of the chip at the end is more correct (important factors for the tool wear and surface quality). Fig. 3 shows this situation.

**Table 2.** Performance indices for the experiments

| Controller | $ITAE$ | $ITSE$ | $IT^2SE$ | $Ovt(\%)$ |
|---|---|---|---|---|
| Without Control | 3.29 | 0.45 | 1.92 | 17.34 |
| ANFIS-IMC | 2.04 | 0.20 | 1.00 | 7.23 |
| NFI-IMC | 1.50 | 0.18 | 0.55 | 6.22 |

Therefore, for all the reason mentioned above, this experimental test serves to conclude that NFI-IMC strategy outperforms the ANFIS-IMC approach.

## 6 Conclusions

From the best of authors' knowledge, the NFI paradigm is applied for the first time to process control. The transductive NFI system is capable of creating local models of the high-performance drilling process for each new input into the control system. With the NFI-IMC system, the material removal rate is successfully increased. In addition, the good quality of the transient response and the overshoot free response contribute from the industrial viewpoint to get more out of the tool's useful life while increasing the material removal rate.

Thus it has been demonstrated that NFI is a simple, fast, precise, computationally viable tool for modeling manufacturing processes. The use of transductive modeling techniques in processes such as drilling proves to be beneficial. Another advantage of NFI is that it makes acceptable predictions with a very small number of data (e.g., the ANFIS-IMC system uses twice as many data in its training set).

The method developed herein is completely valid for control in other conditions. All that needs to be done is to enter in the training set experimental data under the required working conditions and adjust the parameters of the algorithms to match the required working conditions. In this paper a viable application of Ethernet in real-time networked process control is shown and is profiled as a low-cost solution.

## References

1. Mitra, S., Hayashi, Y.: Neuro-fuzzy rule generation: survey in soft computing framework. IEEE Transactions on Neural Networks 11, 748–768 (2000)
2. Keller, J.M., Hunt, D.J.: Incorporating fuzzy membership functions into the perceptron algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI 7, 693–699 (1985)
3. Hunt, K.J., Sbarbaro, D.: Neural networks for nonlinear internal model control. In: IEE Proceedings D: Control Theory and Applications, vol. 138, pp. 431–438 (1991)
4. Jang, J.-S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics 23, 665–685 (1993)

5. Kim, J., Kasabov, N.: HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. Neural Networks 12, 1301–1319 (1999)
6. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, Inc., New York (1998)
7. Atkeson, C.G.: Using local models to control movement. In: Advances in Neural Information Processing Systems, vol. 2, pp. 316–323 (1990)
8. Schaal, S., Atkeson, C.G.: Robot juggling: implementation of memory-based learning. IEEE Control Systems Magazine 14, 57–71 (1994)
9. Li, H.X., Deng, H.: An approximate internal model-based neural control for unknown nonlinear discrete processes. IEEE Transactions on Neural Networks 17, 659–670 (2006)
10. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally Weighted Learning for Control. Artificial Intelligence Review 11, 75–113 (1997)
11. Song, Q., Kasabov, N.K.: NFI: A neuro-fuzzy inference method for transductive reasoning. IEEE Transactions on Fuzzy Systems 13, 799–808 (2005)
12. Gajate, A., Haber, R.E.: Internal Model Control Based on a Neurofuzzy System for Network Applications. A Case Study on the High-Performance Drilling Process. IEEE Transactions on Automation Science and Engineering 6, 367–372 (2009)

# Sentiment Classification with Support Vector Machines and Multiple Kernel Functions

Tanasanee Phienthrakul[1], Boonserm Kijsirikul[2], Hiroya Takamura[3],
and Manabu Okumura[3]

[1] Department of Computer Engineering, Faculty of Engineering, Mahidol University,
25/25 Phutthamonthon, Salaya, NakornPathom, 73170 Thailand
[2] Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University,
254 Phyathai Road, Patumwan, Bangkok 10330 Thailand
[3] Precision and Intelligence Laboratory, Advanced Information Processing Division, Tokyo
Institute of Technology, 2459 Nagatsuta, Midori-ku, Yokohama, 226-8503 Japan
tanasanee@yahoo.com, boonserm,k@chula.ac.th,
takamura@pi.titech.ac.jp, oku@pi.titech.ac.jp

**Abstract.** Support vector machine (SVM) is a learning technique that performs well on sentiment classification. The performance of SVM depends on the used kernel function. Hence, if the suitable kernel is chosen, the efficiency of classification should be improved. There are many approaches to define a new kernel function. Non-negative linear combination of multiple kernels is an alternative, and the performance of sentiment classification can be enhanced when the suitable kernels are combined. In this paper, we analyze and compare various non-negative linear combination kernels. These kernels are applied on product reviews to determine whether a review is positive or negative. The results show that the performance of the combination kernels that outperforms the single kernels.

**Keywords:** Sentiment Classification, Support Vector Machine, Evolutionary Strategy.

## 1 Introduction

Sentiment classification has been a focus of recent research. It has been applied on different domains such as movie reviews, product reviews, and customer feedback reviews [1]. The most basic task in sentiment classification is to classify a document into positive or negative sentiment. The difficulty of sentiment classification is the context-dependency of the sentiments of linguistic expressions. For example, negation words such as "not" or "never" shift the sentiment. A positive statement becomes negative when it is subcategorized by a verb "doubt". Although we could use n-grams (continuous n words) as features in order to handle such shifts, dependency between two words with a long distance cannot be captured by n-grams. Instead of using n-grams, we utilize higher-degree kernel functions, which can automatically take into account the conjunctions of features.

Support vector machine (SVM) is a kernel method that is successful in many different fields. There are many pieces of research work that applied SVM on sentiment

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 583–592, 2009.

classification problems. The results show that SVM yields the best result when it is compared with other approaches such as in [2] and [3]. SVM classifier can be trained using a large number of features [2]. Besides, the learning process of SVM optimizes the margin between two classes on feature space by using the kernel techniques. We however confront another difficulty that it is unknown which kernel function is suitable for this task. In spite of our intuition that feature combinations will capture the context-dependency of sentiment, some researchers have reported that higher-degree kernels only degraded the classification performance [4], [5]. Meanwhile, there is a report that higher-degree kernels did improve the classification performance [6]. This lead us to the idea of using the non-negative linear combination of multiple kernels with adjusted weight parameters.

In using the combination, we need to determine the weight on each kernel in the combination. In order to obtain the suitable weights, we propose to use an evolutionary strategy. The objective function is an important part in evolutionary algorithms, and there are many ways to measure the fitness of the parameters. Training accuracy or training error is the basic function that can be used for evaluating the parameters. Although this function is very easy to calculate, it may cause the overfit to training data. Hence, we propose to estimate the generalization performance of the learning model by the bound that can be derived from the stability property. It is a tight bound, and can be a good criterion for evaluating the parameters in the evolutionary process.

In this paper, we propose the non-negative linear combination kernels and their normalization in Section 2. There parameters are adjusted by the evolutionary strategies that is illustrated in Section 3. Our approach is tested on sentiment classification and the results are reported in Section 4, and the last section is the conclusion.

## 2 Non-negative Linear Combination Kernels

The support vector machine is a classifier which finds the optimal separating hyperplane in terms of some generalization criterion [7]. In the simple pattern recognitions, SVM uses a linear separating hyperplane to create a classifier with the maximum margin [8]. In soft margin SVM, the width of margin can be controlled by a regularization parameter $C$ [8]. The constant $C > 0$ determines the trade-off between margin maximization and training error minimization [9]. For non-linear problems, there is an important technique, called *kernel method*, which enables these machines to produce complex nonlinear boundaries inside the original space. This is performed by mapping the input space into a higher dimensional feature space through a mapping function $\Phi$ and separating there [9]. However, in SVM, it is not necessary to know the explicit form of $\Phi$ . Only the inner product in the feature space, $K(x, y) = \Phi(x) \cdot \Phi(y)$ called *kernel function*, must be defined.

There are many functions that can be used as the kernel in SVM. These kernels are suitable for some problems, and they must be chosen for the tasks under consideration by hand or using prior knowledge [8]. In this paper, we take an interest in four kernel functions i.e. linear, polynomial, RBF, and sigmoid kernels. These kernel functions are shown in Table 1. In order to obtain a better result, the non-negative linear combination of kernels is proposed for SVM on sentiment classification. The analytic expression of this kernel is the following:

$$\mathrm{K}(x, y) = \sum_{i=1}^{n} a_i K_i(x, y) , \tag{1}$$

where $n$ is the number of sub-kernels, $a_i \geq 0$ for $i = 1,...,n$ are the arbitrary non-negative weighting constants, and $K_i(x, y)$ for $i = 1,...,n$ are the sub-kernels.

**Table 1.** Common Kernel Functions

| Kernel | Formula |
|---|---|
| Linear | $K(x, y) = x \cdot y$ |
| Polynomial | $K(x, y) = (x \cdot y + c)^d$ |
| Gaussian RBF | $K(x, y) = \exp(-\gamma \lVert x - y \rVert^2)$ |
| Sigmoid* | $K(x, y) = \tanh(x \cdot y + c)$ |

*This kernel may not be Mercer's kernel.

The linear, polynomial, and RBF kernels are the well-known Mercer's kernels. Therefore, the non-negative linear combination of these kernels still corresponds to the Mercer's theorem. Although it is known that the sigmoid function may not be Mercer's kernel, the sigmoid function is quite popular for support vector machines. Therefore, the kernel functions that will be combined to create a new kernel function are chosen from these four popular kernels. The examples of non-negative linear combination kernels are shown in (2), (3), and (4). These kernels are more flexible as it has more adjustable parameters.

$$\mathrm{K}(x, y) = a_1 K_{Linear}(x, y) + a_2 K_{Poly}(x, y) + a_3 K_{RBF}(x, y) + a_4 K_{Sigmoid}(x, y) \tag{2}$$

$$\mathrm{K}(x, y) = a_1 K_{Linear}(x, y) + a_2 K_{Poly}(x, y) + a_3 K_{RBF}(x, y) \tag{3}$$

$$\mathrm{K}(x, y) = a_1 K_{Linear}(x, y) + a_2 K_{Poly}(x, y) \tag{4}$$

Besides, the non-negative linear combinations of polynomial and RBF kernels are proposed. With these two kind of kernel functions, there are three possible ways of non-negative linear combinations in order to combined them, i.e., (1) the non-negative linear combination of several polynomial kernels at different degree, (2) the non-negative linear combination of several RBF kernels at different scale, and (3) the non-negative linear combination of both polynomial and RBF kernels at different parameters.

Then, normalization is an important pre-processing [10]. Normalization in feature space is not applied directly on the input vector, but it can be seen as a kernel interpretation of the preprocessing [11]. This normalization redefines a new kernel function $\tilde{\mathrm{K}}(x, y)$ of SVM. The non-negative linear combination kernels are normalized by

$$\tilde{\mathrm{K}}(x, y) = \frac{\mathrm{K}(x, y)}{\sqrt{\mathrm{K}(x, x)\,\mathrm{K}(y, y)}} . \tag{5}$$

The normalized kernels place the data on a portion of the unit hypersphere in the feature space [10]. Obviously, the equation $\tilde{\mathrm{K}}(x, x) = 1$ holds true.

# 3 Weight Adjustment

In the non-negative linear combination kernels, the weight of each sub-kernel is the adjustable parameters. These parameters can be determined by executing a grid search and measuring some goodness criterion at each point (criteria are discussed in Section 3.4), but this kind of search consumes a lot of time especially when there are multiple adjustable parameters. The evolutionary strategy (ES, [12]) is a method that can efficiently find the optimal parameters; it is based on the principles of adaptive selection found in the natural world. Each generation (iteration) of the ES algorithm takes a population of individuals (potential solutions) and modifies the problem parameters to produce offspring (new solutions) [13]. Only the highest fit individuals (better solutions) survive to produce new generations [13]. This algorithm has been successfully used to solve various types of optimization problems [14]. Hence, we propose to use the evolutionary strategies for choosing the parameters of non-negative linear combination kernels and SVM.

There are several different versions of ES. Nevertheless, we prefer to use the (5+10)-ES where 5 solutions are used to produce 10 new solutions by a recombination method. These new solutions are mutated and evaluated, and only the 5 fittest solutions are selected from 5+10 solutions to be the parents in the next generation. These processes will be repeated until a fixed number of generations have been produced or the acceptance criterion is reached.

## 3.1 Initialization

Let $\vec{v}$ be the non-negative real-value vector of the parameters. The vector $\vec{v}$ has $n+1$ dimensions and it is represented in the form:

$$\vec{v} = (C\ ,\ a_1,\ a_2\ ,\ \dots\ ,\ a_n), \tag{6}$$

where $C$ is the regularization parameter, $a_i$ for $i = 1,\dots,n$ are the weights of each sub-kernel, and $n$ is the number of terms of sub-kernel.

For multiple RBF sub-kernels, the width of RBF will be added into the vector of parameters ( $\vec{v}$ ). They also will be investigated by the evolutionary strategy. The (5+10)-ES algorithm starts with the 0th generation ($t$=0) that selects 5 solutions ( $\vec{v}_1,\dots,\vec{v}_5$ ) and standard deviation $\vec{\sigma} \in R_+^{n+1}$ using randomization. These 5 initial solutions are evaluated to calculate their fitness. Our goal is to find $\vec{v}$ that optimizes the objective function $f(\vec{v})$ that will be carefully designed in Section 3.4.

### 3.2 Recombination

For each generation, the 5 fittest solutions are assigned the probabilities of selection to create new solutions. These fittest solutions are ordered by their objective functions, i.e. $\vec{v}_i$ is fitter than $\vec{v}_{i+1}$. Then, their probabilities are assigned by

$$\text{Prob}(\vec{v}_i) = \frac{\mu-(i-1)}{\sum_{j=1}^{\mu} j} = \frac{\mu-(i-1)}{\mu(\mu+1)/2} = \frac{2}{\mu}\left(1-\frac{i}{\mu+1}\right) \tag{7}$$

for $i = 1, 2, \ldots, \mu$, when $\mu$ is the number of fittest solutions. In this case, $\mu$ is equal to 5.

After that, any 2 solutions are randomly selected from the conventional 5 solutions with their probabilities. Then, the average of this pair of solutions, element by element, is a new solution. This method is called the global intermediary recombination method, and it will be used to create 10 new solutions.

### 3.3 Mutation

The $\vec{v}_i'$ for $i=1,\ldots,10$ are mutated by adding each of them with $\vec{z}$ where

$$\vec{z} = (z_1, z_2, \ldots, z_{n+1}) \tag{8}$$

when $z_i$ is a random value from a normal distribution with zero mean and $\sigma_i^2$ variation.

$$\begin{gathered} mutate(\vec{v}) = (C+z_1, a_1+z_2, a_2+z_3, \ldots, a_n+z_{n+1}) \\ z_i \sim N_i(0,\sigma_i^2) \end{gathered} \tag{9}$$

Moreover, in each generation, the standard deviation will be adjusted by (14) when $\tau$ is an arbitrary constant.

$$\begin{gathered} mutate_{\sigma}(\vec{\sigma}) = (\sigma_1 \cdot e^{z_1}, \sigma_2 \cdot e^{z_2}, \ldots, \sigma_{n+1} \cdot e^{z_{n+1}}) \\ z_i \sim N_i(0,\tau^2) \end{gathered} \tag{10}$$

### 3.4 Objective Function

In general, training error can be used as the objective function in the evolutionary processes. However, this function may cause the overfit to training data. Sometimes, data contain a lot of noise, and thus if the model fits these noisy data, the learned concept may be wrong. Hence, this paper proposed to use the bound of generalization error that is derived from the assumption of stability. The concept of stability was proposed by Bousquet and Elisseeff [15]. They defined the notions of stability for learning algorithms and showed how to use the notions to derive generalization error bounds [15]. In this work, the stability of soft margin SVM classification is applied to be the objective function in evolutionary process in order to avoid the overfitting problem.

**Proposition.** (Stability of soft margin SVM classification) Let $(x_1, y_1)$, $(x_2, y_2), \dots,$ $(x_m, y_m)$ be the training data where $x_j \in R^N$ is a sample data and $y_i \in \{-1, 1\}$ is its label. Assume $K(\cdot)$ is a bounded kernel, that is $K(x_i, x_j) \le \kappa^2$. The bound with probability at least $1-\delta$ over the sample of size $m$ is

$$R \le R_{emp} + \frac{\kappa^2}{\lambda m} + \left(1 + \frac{2\kappa^2}{\lambda}\right)\sqrt{\frac{\ln(1/\delta)}{2m}}\,, \tag{11}$$

where $R$ is the risk or generalization error, $R_{emp}$ is called the empirical error, and $\lambda$ is the regularization parameter of SVM ( $\lambda = 1/C$ ).

The expressions in the right-hand side of (11) are used as the objective function to evaluate parameters of SVM and kernel function. The bound of kernel function ( $\kappa^2$ ) can be estimated when the parameters of the kernel are assigned for each individual vector ( $\vec{v}$ ). We presume that a set of suitable parameters should provide a lower bound of risk.

## 4 Sentiment Classification

We used a dataset of product reviews, which was provided by Bing Liu[1] [16]. This dataset contains sentences used in product reviews collected from the internet and assigned with a sentiment tag: positive or negative. The dataset contains 1,700 sentiment sentences: 1,067 positive and 633 negative sentences. Methods were evaluated by 5-fold cross-validation. The SVM classifiers were trained by using unigrams (single words) as features. The evolutionary strategies were used to find the optimal parameters. The value of $\tau$ in evaluation process of these experiments is 1.0. The weights of combination $(a_i)$, and the regularization parameter ( $C$ ) were real numbers between 0.0 and 10.0. These parameters were inspected within 1,000 generations of ES. The non-negative linear combination kernels for sentiment classification were compared in terms of the average test accuracy. The single kernel functions, i.e., linear kernel, polynomial kernel at different degree, RBF at different scale, and sigmoid kernel are the baselines. The average accuracy values of SVM with single kernel functions are shown in Table 2.

From Table 2, we can see that the linear kernel yielded a good accuracy whereas the sigmoid kernel did not. For polynomial and RBF kernels, the accuracy of sentiment classification is decreased when the degree of polynomial or the width of RBF is increased. The graphs of polynomial and RBF kernels when we varied the parameters are shown in Fig. 1, respectively.

From those graphs, we found that the polynomial at degree 2 yields the result that is better than the other degree. Therefore, we will use degree 2 of polynomial kernel to combine with other kernel functions in the non-negative linear combination kernel.

[1] The dataset is available at http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

**Table 2.** The Average Accuracy of SVM with Single Kernel Functions

| Kernel | Average Accuracy | Kernel | Average Accuracy |
|---|---|---|---|
| Linear | 77.7093 | Sigmoid | 56.0037 |
| Polynomial (d = 2) | 69.7711 | RBF ( $\gamma$ = 0.0005) | 77.8266 |
| Polynomial (d = 3) | 65.0086 | RBF ( $\gamma$ = 0.005) | 78.1216 |
| Polynomial (d = 4) | 63.4780 | RBF ( $\gamma$ = 0.05) | 75.3579 |
| Polynomial (d = 5) | 62.7721 | RBF ( $\gamma$ = 0.5) | 62.6546 |
| Polynomial (d = 6) | 62.7723 | RBF ( $\gamma$ = 5.0) | 62.6546 |

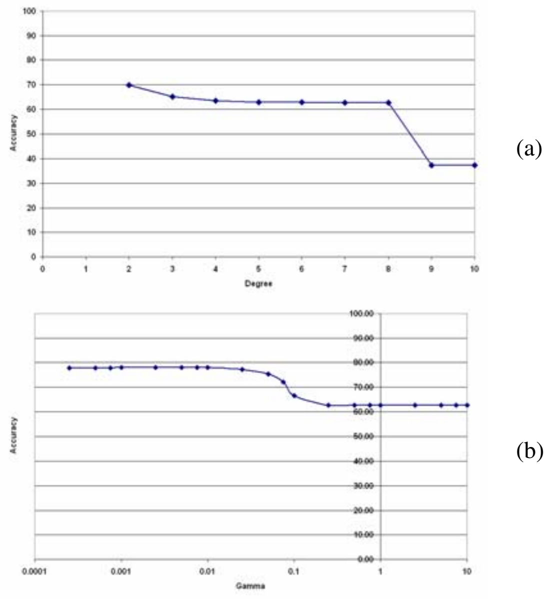


**Fig. 1.** The accuracy of sentiment classification at different parameters, (a) on Polynomial kernels and (b) on RBF kernels

In the same way, the accuracy of RBF kernel at the width 0.005 is better than those of the other parameters. Hence, the RBF kernel at the width 0.005 will be used for testing on the combined kernels. The average accuracy of SVM with non-negative linear combination kernels is shown in Table. 3.

We tested the SVM with all possible linear combination of these 4 sub-kernels (linear, polynomial, RBF, and sigmoid), and found that the non-negative linear combination of linear and polynomial kernels yields the best result. Although the linear combination of three or four sub-kernels yield more average accuracy on training, the average accuracy on test is not good. This means that the performance of sentiment classification will be improved when the suitable kernels are combined. If we choose

**Table 3.** The Average Accuracy of SVM with Non-Negative Linear Combination Kernels

| Kernel | Average<br>Training Accuracy | Average<br>Test Accuracy |
|---|---|---|
| Linear + Poly (2) + RBF (0.005) + Sigmoid | 99.2218 | 69.4782 |
| Linear + Poly(2) + RBF(0.005) | 99.7647 | 70.3580 |
| Linear + Poly(2) + Sigmoid | 99.8236 | 69.7670 |
| Linear + RBF(0.005) + Sigmoid | **99.8530** | 70.2391 |
| Poly(2) + RBF(0.005) + Sigmoid | 99.7059 | 68.1829 |
| Linear + RBF(0.005) | 98.7355 | 78.7122 |
| Linear + Poly(2) | 98.0881 | **79.1828** |

the unsuitable kernel, the accuracy may be decreased or the classification model may overfit training data.

Then, the non-negative linear combination of several polynomial kernels at different degree, the non-negative linear combination of several RBF kernels at different scale, and the non-negative linear combination of both polynomial and RBF kernels at different parameters are validated. The number of terms of sub-kernel was fixed as 10. The degree of polynomial sub-kernels are 1, 2, …, 10 for multiple polynomial kernels. For multiple polynomial and RBF kernels, 5 terms of polynomial sub-kernels and 5 terms of RBF sub-kernels are used. The weights of combination $(a_i)$, the widths of RBF kernels $(\gamma_i)$, and the regularization parameter $(C)$ were real numbers between 0.0 and 10.0. The average accuracy of SVM with non-negative linear combination of multiple polynomial and RBF kernels is shown in Table. 4. These results are compared with the single kernel functions.

**Table 4.** The Average Accuracy of SVM with Non-Negative Linear Combination of Multiple Polynomial and RBF Kernels

| Kernel | Average<br>Training Accuracy | Average<br>Test Accuracy |
|---|---|---|
| Linear | 88.3533 | 77.7093 |
| Poly(2) | 85.7943 | 69.7711 |
| RBF(0.005) | 89.6472 | 78.1216 |
| Multiple Polynomial Kernels | 98.0881 | **79.1828** |
| Multiple RBF Kernels | 95.0294 | 73.1805 |
| Multiple Polynomial and RBF Kernels | **98.7355** | 78.7122 |

The sentiment classification was tested by the SVM with all 3 kinds of non-negative linear combination on multiple polynomial and RBF kernels. We found that the average accuracy on sentiment classification can be enhanced by the combined kernel.

The non-negative linear combination of multiple polynomial kernels yielded the best result on testing. Although the linear combination of multiple RBF kernels did not yield the best result, its accuracy was better than single RBF kernel. For the combination of both polynomial and RBF kernels, it yielded the best training accuracy, but its accuracy on testing is lower than the combination of multiple polynomial kernels. This means that although we try to avoid the overfitting problem by using the stability objective function in evolutionary process, the overfiitng problem still can be occurred by a more flexible kernel.

## 5 Conclusion

The non-negative linear combination kernels for SVM were proposed and applied on the sentiment classification problem. This kernel function was more flexible, and there were some adjustable parameters (the weights of combination). The evolutionary strategies were used for adjusting these parameters. In order to avoid the overfitting problem, the stability of soft margin SVM classification was considered to be the objective function in evolutionary process.

The experimental results showed the ability of the proposed method through the average accuracy on 5-fold cross-validation on the sentiment classification problem. The non-negative linear combination kernels yielded the classification results that were better than the single RBF kernels when the suitable kernels were combined. If many sub-kernels were combined, it maybe occur the overfitting problem. However, there are the other combination methods that maybe improve the efficiency of sentiment classification, which we will be investigated in the near future.

## Acknowledgement

The authors acknowledge the financial support provided by the Royal Thai Government Scholarship (Mahidol University) and the Thailand Research Fund.

## References

1. Kennedy, A., Inkpen, D.: Sentiment Classification of Movie Reviews using Contextual Valence Shifters. Computational Intelligence 22(2), 110–125 (2006)
2. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79–86 (2002)
3. Li, J., Sun, M.: Experimental Study on Sentiment Classification of Chinese Review using Machine Learning Techniques. In: International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE 2007), Beijing, pp. 393–400 (2007)
4. Mullen, T., Collier, N.: Sentiment Analysis using Support Vector Machines with Divers Information Sources. In: Proceedings of EMNLP (2004)
5. Li, S., Zong, C., Wang, X.: Sentiment Classification through Combining Classifiers with Multiple Feature Sets. In: International Conference on Natural Language Processing and Knowledge Engineering, 2007 (NLP-KE 2007), pp. 135–140 (2007)

6. Okanohara, D., Tsujii, J.: Assigning Polarity Scores to Reviews Using Machine Learning Techniques. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, pp. 314–325. Springer, Heidelberg (2005)
7. Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons, New York (1998)
8. Kecman, V.: Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models. MIT Press, London (2001)
9. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, London (2002)
10. Graf, A., Borer, S.: Normalization in Support Vector Machines. In: Radig, B., Florczyk, S. (eds.) DAGM 2001. LNCS, vol. 2191, pp. 277–282. Springer, Heidelberg (2001)
11. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
12. Beyer, H.-G., Schwefel, H.-P.: Evolution strategies: A comprehensive introduction. Natural Computing 1(1), 3–52 (2002)
13. de Doncker, E., Gupta, A., Greenwood, G.: Adaptive Integration Using Evolutionary Strategies. In: Proceedings of 3rd International Conference on High Performance Computing, pp. 94–99 (1996)
14. Fogel, D.B.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. IEEE Press, Piscataway (1995)
15. Bousquet, O., Elisseeff, A.: Stability and Generalization. Journal of Machine Learning Research 2, 499–526 (2002)
16. Hu, M., Liu, B.: Mining and Summarizing Customer Reviews. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004), Seattle, Washington, USA, August 22-25 (2004)

# Improving the Performance of Fuzzy ARTMAP with Hybrid Evolutionary Programming: An Experimental Study

Shing Chiang Tan[1] and Chee Peng Lim[2]

[1] Faculty of Information Science & Technology, Multimedia University
Melaka Campus, Jalan Ayer Keroh Lama, Bukit Beruang, 75450 Melaka, Malaysia
sctan@mmu.edu.my
[2] School of Electrical & Electronic Engineering, University of Science Malaysia
Engineering Campus, 14300 Nibong Tebal, Penang, Malaysia
cplim@eng.usm.my

**Abstract.** This paper presents an evolutionary artificial neural network (EANN) that combines the operations of Fuzzy ARTMAP (FAM) and a Hybrid Evolutionary Programming (HEP) model in a sequential manner. The proposed FAM-HEP network, which harnesses the advantages of FAM and HEP, is able to construct its network architecture autonomously, and to perform learning and evolutionary search and adaptation. In order to evaluate the effectiveness of the proposed FAM-HEP network, an experimental study using benchmark data sets is conducted. The performance of FAM-HEP is analyzed, and the results are compared with those of FAM-EP and FAM. Overall, FAM-HEP outperforms FAM-EP and FAM. The study also reveals the potential of FAM-HEP as an innovative EANN model for undertaking pattern classification problems.

**Keywords:** Fuzzy ARTMAP, Hybrid Evolutionary Programming, Pattern Classification.

## 1 Introduction

Artificial Neural Networks (ANNs), which are inspired by the biological nervous systems, consist of many processing elements operating in parallel to perform computation (DARPA, 1988). Over the years, many research efforts have been devoted to applying Evolutionary Algorithms (EAs) to evolve different aspects of ANNs. EAs are computational models that are inspired by natural evolution and adaptation. The combination between ANNs and EAs results in a special class of ANNs known as evolutionary ANNs (EANNs). Yao (1999) and Buchtala et al. (2005) provide a good review on EANNs. EANNs capitalize evolution as another fundamental form of adaptation in addition to learning (Yao, 1999). It is these adaptation and learning properties that enable EANNs to perform well in dynamic environments.

From the literature, feedforward ANNs (Yao, 1999) and radial basis function networks (Buchtala et al., 2005) are among the popular ANNs for combination with EAs. However, less attention is paid to devise EANNs by utilizing the Adaptive Resonance

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 593–600, 2009.

Theory (ART) family of networks (Grossberg, 1976, Carpenter and Grossberg, 1987). The ART networks have a distinctive feature as compared with most feedforward ANNs, i.e., the ability to overcome the stability-plasticity dilemma (Carpenter and Grossberg, 1987). The network is *plastic* enough to learn new information from new data samples, and, at the same time, is *stable* enough to preserve previously learned information from corruption upon acquiring new information. From the viewpoint of classification, this capability means that the network can continue to learn knowledge safely *in situ* while providing useful predictions. As such, we propose an ART-based EANN for tackling pattern classification tasks in this paper. In particular, the Fuzzy ARTMAP (FAM) network, a supervised ART model (Carpenter et al., 1992), is utilized to construct an EANN. A salient feature of FAM is that it combines fuzzy set theory and the stability-plasticity of ART into a common framework, and it is also able to handle both analog and binary input patterns.

The performance of FAM can be improved if its dynamics are adapted with a search algorithm provided by the EA. For many researchers, Evolutionary Programming (EP) is deemed as a feasible EA for evolving ANNs (Palmes et al., 2005). A significant advantage of EP over other EAs, for instance the Genetic Algorithm (GA), is that EP is not affected by the *deceptive mapping* problem (Goldberg, 1989) or the permutation problem (Yao, 1999), which causes inefficiency in the evolution process of ANNs. The reason is, EP, which solely uses the mutation operator, allows adaptation in the evolutionary process to be conducted without exchanging information among chromosomes (networks). However, while EP is capable of identifying the basins of optima during search, it usually is unable to exploit the basins effectively to reach the global optimum. Martínez-Estudillo et al. (2006) argued that the lack of precision of the GA search for solutions can be overcome by incorporating the GA with a local search algorithm. The integration of the GA with a local fine-tuning algorithm is called a Hybrid GA. In this paper, the proposed model that fuses FAM and Hybrid EP (HEP) together is also inspired from such idea. The features of both FAM and HEP cross-fertilize each other to form a FAM-HEP network. The FAM network, which possesses the ability to overcome the stability-plasticity dilemma, is able to provide information directly from data samples for constituting an encoding scheme as required by HEP. Meanwhile HEP, which is a kind of evolution incorporating local refinement, is able to provide robust search coverage on provision of the information encoded by FAM. Hence, the proposed FAM-HEP network can undergo adaptive learning through FAM, and the network dynamics can be further enhanced with HEP.

The organization of this paper is as follows. The proposed FAM-HEP model is explained in details in section 2. In section 3, the classification ability of FAM-HEP is evaluated using benchmark data sets from the UCI Machine Learning Repository (Asuncion and Newman, 2007). The results of FAM-HEP are compared with those without a local search algorithm (i.e., FAM-EP) and FAM. Concluding remarks and suggestions for further work are included in section 4.

## 2 Fuzzy ARTMAP and Hybrid Evolutionary Programming

In this section, FAM, EP, and local search algorithm are briefly described. For details, readers can refer to Carpenter et al. (1992) (for FAM), Yao (1999) (for EP), and

Baskar et al. (2001) (for local search algorithm). The proposed FAM-HEP is also described.

### 2.1 Fuzzy ARTMAP (FAM)

The architecture of FAM comprises two Fuzzy ART modules (i.e., ART$_a$ the input module and ART$_b$ the output module) that are linked together through a mapping field $F^{ab}$ (Carpenter et al., 1992). Figure 1 shows a schematic diagram of the FAM architecture. Each fuzzy ART module is composed of three layers of nodes: $F_0$ is the input layer that receives an input pattern; $F_1$ is the layer that is fully connected to $F_2$ by a set of connection weights, $\boldsymbol{w}$ ; $F_2$ is the dynamic layer that encodes the prototypes of input patterns (represented as the nodes), and new nodes can be added into this layer when necessary.

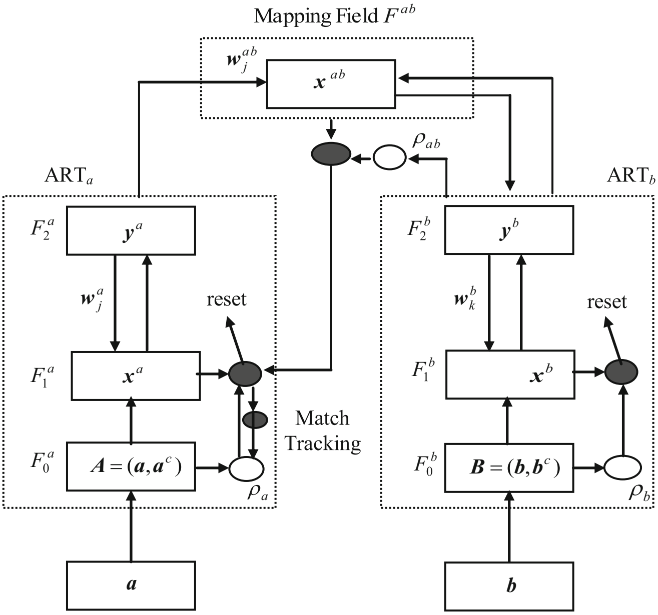


**Fig. 1.** The FAM architecture

The FAM network is able to self-organize and self-stabilize information and network architecture on deployment of a feedforward pass and a feedback pass for pattern matching. The network grows with time, subject to vigilance tests (Carpenter et al., 1992) that are governed by a user-defined vigilance parameter $\overline{\rho} \in [0,1]$, and a mapping field vigilance parameter $\rho_{ab}$. During supervised learning, an input pattern (output class) is presented to ART$_a$ (ART$_b$), respectively. The vigilance test is conducted in each fuzzy ART module to evaluate the similarity between the prototypes (weights of the network nodes) and the input pattern (output class), and to select a winning node. After obtaining a winning node in each fuzzy ART module, resonance is said to occur after all vigilance tests have been satisfied. In this case, a permanent link is established at the mapping field to connect the winning nodes in ART$_a$ and ART$_b$. However, if none of the existing prototypes from either fuzzy ART module can satisfy all vigilance tests, a new node is introduced, and the input pattern is coded as its prototype. Hence, subject to the vigilance criterion, the number of nodes grows with time, and this leads to the formation of an autonomous network architecture for tackling the problem under scrutiny.

### 2.2 Evolutionary Programming (EP)

EP was proposed by Fogel et al. (1966) to simulate intelligent behaviour by means of finite-state machines. Candidate solutions to a problem are considered as a population of finite-state machines, and the whole population undergoes a search and adaptation process through a number of evolution generations. In this regard, new candidate solutions (offspring) are generated from the existing candidate solutions (parents) through mutation. All candidate solutions are evaluated by a fitness function. In EP, the crossover operator is not used. Instead, only the mutation operator is employed to provide variation in the population of candidate solutions. EP undergoes a search and adaptation process in a number generation of evolution. In each generation, the whole population of chromosomes undergoes a cyclic process of mutation, chromosome fitness evaluation and selection. Such a process is repeated until a terminating criterion, e.g., either a fittest chromosome (the chromosome achieves the highest accuracy rate) is found from its population, or the maximum number generation of evolution is reached.

### 2.3 Direct Search Algorithm

The Direct Search (DS) algorithm (Baskar et al., 2001) is a local search algorithm that is integrated with the real-coded GA for finding an optimized solution to the nonlinear constrained problem. In their work, the real-coded GA is employed as a base level search for the candidate solutions that are located in the basins of global optimum. DS, which provides a systematic procedure to reduce the size of the search space, is then applied to refine the candidate solutions that have been obtained by the real-coded GA for finding an exact optimal solution. In this work, while EP is capable of identifying the basin of global optima for ANN solutions, it is usually unable to exploit the basin effectively to reach the exact global optimum solution for the ANN to achieve a better recognition performance. In this case, it is worthwhile to apply a local search algorithm to refine the "coarse" solution that has been found by EP. In this paper, we adopt this idea. Hence, DS is employed for integration with EP (i.e., HEP) to evolve FAM.

### 2.4 Fusion between FAM and HEP

The FAM-HEP model is an EANN that performs adaptation in two modes: the learning mode of ART-based computation; and, the evolution mode of HEP-based search and optimization of the network weights. The adaptation process of the FAM-HEP model can be summarized as follows.

(1) FAM training. The network is trained according to the standard FAM learning algorithm. Upon receiving the supervised training data samples, prototypes of the input features and output classes are formed as a set of weights in ART$_a$ (i.e., $\boldsymbol{w}_a$) and ART$_b$ (i.e., $\boldsymbol{w}_b$), respectively.

(2) The formation of initial population. The weights of ART$_a$ (i.e., $\boldsymbol{w}_a \equiv [\boldsymbol{w}_a^1, \boldsymbol{w}_a^2, \cdots, \boldsymbol{w}_a^j, \cdots \boldsymbol{w}_a^N]$) contain prototypes of the input features that can be used to improve the network classification performance. Each weight vector is concatenated into a string as a Pittsburgh chromosome (Kitano, 1990; Yao, 1999) (i.e., $z_0 = [\boldsymbol{w}_a^1 \boldsymbol{w}_a^2 \cdots \boldsymbol{w}_a^j \cdots \boldsymbol{w}_a^N]$). Based on $z_0$, a finite number of chromosomes are generated to form a population according to (Baskar et al., 2001):

$$z_i = z_0 + \text{RMF}\,.\,\text{rand}(1, n) \tag{1}$$

where $z_i$ is the $i$-th replicated chromosome of $z_0$ (or the $i$-th replicated ART$_a$ from its initial setup in FAM); RMF$\in [0,1]$ (Range Multiplication Factor) is a user-defined scalar; "." represents scalar multiplication, and rand(1,$n$) $\in [0,1]$ is a uniformly distributed random vector, with $n$ representing the length of $z_0$.

(3) Evolving a population of chromosomes using EP. The whole population of chromosomes undergoes a fitness evaluation process, and a global search and adaptation process using a roulette-wheel selection operator and a Gaussian mutation operator (Yao, 1999). The fitness of each chromosome is measured in terms of the network accuracy based on training data. The roulette-wheel selection strategy is adopted so that the chromosomes with higher fitness values are more likely to be chosen to form a new population of chromosomes for evolution in subsequent generation. On the other hand, a mutation probability, $p_m$, is introduced to control the mutation operation. When an arbitrary probability that is lower than $p_m$ occurs, Gaussian mutation is activated to change the chromosome. The global search and adaptation process as in step (3) is repeated until a terminating criterion is met.

(4) Evolving a population of chromosomes using the DS algorithm. The fittest chromosome $z_f$ from step (3) is selected. A population of candidate solutions is generated based on $z_f$ in the solution space. In this case, a systematic procedure as in Baskar et. al (2001) is exercised to reduce the size of the search space with the intention to refine the candidate solutions from step (3) for improving classification performance.

## 3 Experiments

To demonstrate the effectiveness of the proposed FAM-HEP model, a series of experiments was conducted using three benchmark data sets, i.e., Pima Indian Diabetes (PID), Australian Credit Approval (AUS), and Heart (HEA). All these data sets are available from the UCI Machine Learning Repository (Asuncion and Newman, 2007).

The PID, AUS and HEA data sets are binary-class problems with 768, 690, and 270 samples, and have 8, 14, and 13 input attributes, respectively.

In each experiment, the data set was divided into two subsets: training and test sets. The FAM-HEP model was trained using 50% randomly selected data while the remaining data were used for evaluating the network performance. In order to ascertain the stability of the performance, each experiment was repeated eight times with different training data sequences. Unless otherwise stated, FAM-HEP was trained with single epoch and with "default" FAM parameters, i.e., fast learning, $\beta = 1$, and ART$_a$ baseline vigilance parameter, $\bar{\rho}_a = 0.0$, while the HEP parameters were set as follows: the number of replicate chromosomes (i.e., $n$=10), RMF=0.30, mutation probability $p_m = 0.20$, and $T$=$LT$=10. In addition to FAM-HEP, the same experiments using FAM-EP and FAM were conducted for performance comparison purposes. Since FAM-EP used only the first phase of evolutionary adaptation of FAM-HEP, the parameters of FAM-EP were the same as those of FAM-HEP, but without $LT$. As for FAM, the standard FAM algorithm was adopted with the same $\beta$ and $\bar{\rho}_a$ values. Note that FAM was trained using both the single-epoch and multi-epoch schemes. Table 1 summarizes the overall results.

**Table 1.** Results of FAM-based networks (Acc. - accuracy; Stdev. - standard deviation)

| | PID | | AUS | | HEA | |
|---|---|---|---|---|---|---|
| Network Model | Acc. (%) | Stdev. | Acc. (%) | Stdev. | Acc. (%) | Stdev. |
| FAM (single epoch) | 64.36 | 3.35 | 67.61 | 3.58 | 62.97 | 4.46 |
| FAM (multiple epochs) | 66.87 | 1.23 | 78.55 | 1.30 | 71.67 | 1.57 |
| FAM-EP | 68.65 | 2.58 | 72.41 | 4.52 | 69.04 | 4.51 |
| FAM-HEP | 71.59 | 1.54 | 81.62 | 3.04 | 73.70 | 3.01 |

From Table 1, one can observe that FAM-HEP produces higher accuracy rates than its counterpart without local search (i.e., FAM-EP). To further compare the results from the statistical point of view, the bootstrap hypothesis test (Efron, 1979) with a significance level of 0.05 was conducted. One advantage of the bootstrap method is that it does not require the data samples to follow a normal distribution. In the hypothesis test, the null hypothesis states no difference between the test accuracy rate of FAM-EP and FAM-HEP whereas the alternative hypothesis claims that the test accuracy rate of FAM-EP is lower than that of FAM-HEP. The $p$-values of the corresponding tests are listed in Table 2. All $p$-values of the test for the accuracy between FAM-EP and FAM-HEP are smaller than 0.05. This indicates that FAM-HEP perform statistically better (at the 95% confidence level) than FAM-EP.

The performances between FAM networks and their counterparts (FAM-EP and FAM-HEP) are also compared with the same alternative hypothesis aforementioned. From Table 2, we can observe that all $p$-values of the results of FAM (both single- and multi-epoch training) and FAM-HEP are smaller than 0.05. These indicate that both single- and multi-epoch FAM networks are inferior than FAM-HEP. FAM-HEP, which undergoes supervised ART learning in addition to an evolutionary adaptation

**Table 2.** Performance comparison using the bootstrapped hypothesis test. Each entry is a *p*-value, which is computed by the bootstrap method for comparing the test accuracy rates of two selected networks (*X* and *Y*). The alternative hypothesis test claims that the average test accuracy rate of *Y* is lower than that of *X*, whereas the null hypothesis states that the performances of both *X* and *Y* are equal.

| | | *p*-value | | |
|---|---|---|---|---|
| *Y* | *X* | PID | AUS | HEA |
| FAM (single-epoch) | FAM (multi-epoch) | 0.038 | 0.000 | 0.002 |
| FAM (single-epoch) | FAM-EP | 0.007 | 0.010 | 0.008 |
| FAM (single-epoch) | FAM-HEP | 0.000 | 0.000 | 0.000 |
| FAM (multi-epoch) | FAM-EP | 0.044 | 0.996 | 0.927 |
| FAM (multi-epoch) | FAM-HEP | 0.000 | 0.009 | 0.043 |
| FAM-EP | FAM-HEP | 0.006 | 0.000 | 0.008 |

process combining both global and local searches for identifying the coarse solution in the basin of solution space and thereafter fine-tuning the solution, manages to give improved classification performances as compared with those of the FAM networks.

By comparing the results between FAM-EP and FAM, it can be seen that, on one hand, FAM-EP performs better than single-epoch FAM; on the other hand, it is inferior to multi-epoch FAM. In this case, FAM-EP, which undergoes global search and adaptation (but without local search), does not always outperform FAM. The may be owing to the suboptimal solutions in the feasible solution space found by FAM-EP. Thus, FAM-HEP, which evolves using both global and local search algorithms, is a better EANN solution for undertaking pattern classification tasks.

## 4 Summary

In this paper, a novel EANN which integrates FAM and HEP into a common framework has been proposed. The applicability of FAM-HEP to pattern classification tasks has been evaluated using three benchmark data sets from the UCI Machine Learning Repository. The results have positively indicated that FAM-HEP is able to yield better classification performances than those of FAM-EP and FAM. This is owing to the use of both global and local search mechanisms to evolve and adapt the prototype weights coded in FAM, which has resulted in improvements in the classification performances.

As part of future work, more experiments with different data sets from real-world applications can be conducted to further ascertain the effectiveness of FAM-HEP as a usable and useful pattern classification system. New variants of FAM-HEP can also be developed by equipping the classifier with different global and/or local search and adaptation algorithms. In addition, the HEP model can be applied to other ANN-based classifiers. All these research activities aim to devise an autonomous, self-organized and self-adaptive intelligent system that is capable of achieving good performances in dealing with real-world pattern classification problems.

## References

Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. School of Information and Computer Science. University of California, Irvine (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html

Baskar, S., Subraraj, P., Rao, M.V.C.: Performance of hybrid real coded genetic algorithms. International Journal of Computational Engineering Science 2, 583–601 (2001)

Buchtala, O., Klimek, M., Sick, B.: Evolutionary optimization of radial basis function classifiers for data mining applications. IEEE Trans. Syst., Man, Cybernet. B 35, 928–947 (2005)

Carpenter, G.A., Grossberg, S.: A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics and Image Processing 37, 54–115 (1987)

Carpenter, G.A., Grossberg, S., Rosen, D.: Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks 4, 759–771 (1991)

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J., Rosen, D.: Fuzzy ARTMAP: A neural network architecture for incremental learning of analog multidimensional maps. IEEE Trans. Neural Networks 3, 698–713 (1992)

DARPA Neural Network Study. AFCEA Press, Fairfax (1988)

Efron, B.: Bootstrap methods: another look at the Jackknife. The Annals of Statistics 7, 1–26 (1979)

Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence through Simulated Evolution. John Wiley, New York (1966)

Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley, Reading (1989)

Grossberg, S.: Adaptive pattern recognition and universal recoding ii: feedback, expectation, olfaction, and illusions. Biological Cybernetics 23, 187–202 (1976)

Kitano, H.: Designing neural networks using genetic algorithms with graph generation system. Complex Syst. 4, 461–476 (1990)

Martínez-Estudillo, A.C., Hervas-Martinez, C., Martinez-Estudillo, F.J., Garcia-Pedrajas, N.: Hybridization of evolutionary algorithms and local search by means of a clustering method. IEEE Transactions on Systems, Man, and Cybernetics B 36, 534–545 (2006)

Palmes, P.P., Hayasaka, T., Usui, S.: Mutation-based genetic neural network. IEEE Trans. Neural Netw. 16, 587–600 (2005)

Yao, X.: Evolving artificial neural networks. Proc. IEEE 87, 1423–1447 (1999)

# "Dead" Chromosomes and Their Elimination in the Neuro-Genetic Stock Index Prediction System

Jacek Mańdziuk and Marcin Jaruszewicz

Warsaw University of Technology, Faculty of Mathematics and Information Science,
Plac Politechniki 1, 00-661 Warsaw, Poland
mandziuk@mini.pw.edu.pl
jaruszewicz@data.pl
http://www.mini.pw.edu.pl/~mandziuk/

**Abstract.** This paper presents a method for a short-term stock index prediction. The source data comes from the German Stock Exchange (being the target market) and two other markets (Tokyo Stock Exchange and New York Stock Exchange) together with EUR/USD and USD/JPY exchange rates. Neural networks supported by a genetic algorithm (GA) are used as the prediction engine. Except for promising numerical results attained by the system the special focus in the paper is on the problem of elimination of *dead* chromosomes, i.e. the ones which cannot be properly assessed.

## 1 Introduction

Computational Intelligence (CI) methods are widely used for stock market prediction. Most of them, however, require specific set up of steering parameters and suitable selection of the input data. Both these issues can be effectively approached with the help of genetic algorithms (GAs). For example, in [1] the authors propose a system for stock market prediction relying on neural networks and fundamental analysis, where the selection of variables is performed by the GA. Alternatively the topology of the network or its internal parameters can be determined by the GA [2]. GAs are also applied to the task of generation of optimal prediction rules/models [3,4] or transformation of the knowledge hidden inside neural network's weights to a self-explanatory set of rules [5].

The prediction system discussed in this paper is a suitable combination of the computational power of neural networks with flexibility and optimization capabilities of genetic algorithms. The goal is to predict the closing value of German Stock Exchange (GSE) index DAX for the next day. The approach relies solely on technical analysis data. Except for the target market the data from two other international markets namely American New York Stock Exchange (NYSE) with DJIA index and Tokyo Stock Exchange (TSE) with NIKKEI 225 (NIKKEI) index is also considered, together with exchange rates of EUR/USD and USD/JPY.

The most suitable set of input variables is chosen by the GA and validated by a simplified neural network training and testing procedure. Due to high volatility of mutual dependencies between input variables the set chosen by the GA is only used for 5 trading days. After this period a new set of variables is chosen for the next 5 trading days, and so on.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 601–610, 2009.

Proposed solution combines the power of the GA in a wide and parallel search with the intrinsic ability of neural networks to discover relations within input data. The system works without human intervention choosing desired input variables autonomously from a large number of possibilities. In independent prediction steps variables are selected with noticeable sense. The existence of characteristic, repeatable patterns of variables within subsequent 5-day windows can be observed.

Whilst the GA was probing different sets of variables several interesting observations were made. For example, the existence of chromosomes in a population, that coded neural networks unable to learn was discovered. Such chromosomes were called *dead*. They were present especially in early iterations of the GA. With appropriate choice of mutation conditions the algorithm tended to consequently eliminate dead chromosomes.

The reminder of this paper is organized as follows: in the next section an overview of the system is presented, section 3 summarizes experimental results published in part in previous papers and section 4 presents and discusses the problem of dead chromosomes in the population. Conclusions are placed in the last section.

## 2 An Overview of the System

The first step of the prediction algorithm is initial data pre-selection. It starts with analysis of plain values of an index, proceeds with defining averages and oscillators, and ends with application of pattern extraction techniques. Since the goal is defined as a one-day prediction, fundamental analysis is not very helpful [6]. Instead, the main focus is on technical analysis and its indicators. Another possibility would be the use of portfolio management methods, which also proved to be efficient in short-term prediction [4,7]. The source data is the opening, highest, lowest and closing values of the index of the selected stock markets in subsequent days. This data serves as the basis for further transformations, e.g. the percentage change of opening value through the last 5, 10 and 20 days, the last percentage change of closing value or averages of opening values through the last 5, 10 and 20 days.

The above variables are intuitively useful for prediction. They provide simple presentation of past values in a compressed way. Especially moving averages allow the algorithm to omit sudden local changes when looking for prediction rules. The predefined periods for averages and changes (5, 10, 20 days) were chosen based on preliminary simulations and suggestions from the literature [8]. Generally speaking these periods need to be adjusted to the assumed prediction horizon (which is equal to one day in this paper). The same time periods are also used in more complex transformations, e.g. in definitions of the oscillators.

A chart of index values with corresponding changes in the period April 7, 2004 – August 26, 2004 is presented in Fig. 1. Sudden changes in different directions are noticeable. On the other hand, general rises and falls of trend are also clearly visible.

On the same basis the oscillators, known in technical analysis, are calculated [6]. The following oscillators are chosen in our experiment: MACD; Williams; Two Averages; Impet; RSI; ROC; SO and FSO. Except for raw values, the buy/sell signals generated by the oscillators are also considered.

After the pre-processing step a large number of variables are available for the algorithm's selection. A method of selecting sub-optimal, in the current situation, set of

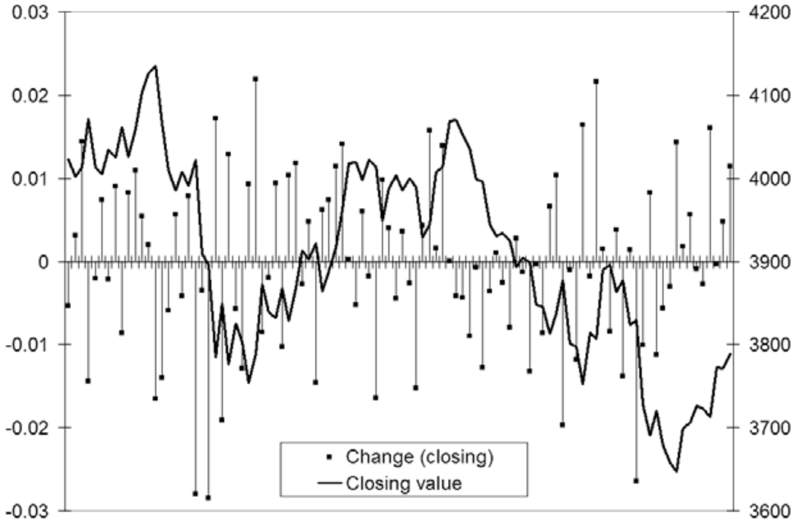


**Fig. 1.** Closing DAX value and its changes in the period 2004/04/07 to 2004/08/26

variables and a neural network-based predictor which uses them as the input are described below. The final selection of the input variables is performed by the GA. Each chromosome codes a set of input variables to be used by a neural network. A fitness function is reverse-proportional to the magnitude of an error made by a trained network on a sample test set. After the GA finishes its work the best fitted chromosome provides a suitable set of variables to be used in the final training. These variables are considered to be a sub-optimal selection for the current 5-day window.

Neural networks form the basis of the proposed prediction system. Feed-forward networks with one hidden layer are chosen based on preliminary experiments. The size of the hidden layer equals the *floor* function of the half of the size of an input layer (e.g. in case of 9 input neurons, the hidden layer is composed of 4 units). One neuron is used in the output layer. Therefore the architecture of each network used in the system during the GA phase or for the final prediction is defined completely by the number of input variables. The weights of such a network are randomly initialized. The method of back propagation with momentum is used for training.

The neural network training process is efficient if the size of the network is relatively small. This goal is typically achieved by adaptation of network's topology [2,9,10,11].

Another possibility of optimization is efficient selection of input variables. In order to select small, but relevant number of variables for a neural network the GA is applied. Due to a large number of available input variables and changing dependencies between them, a selection of a suitable subset of inputs is the main idea of the proposed system. The quality of prediction strictly depends on the effectiveness of the GA. The chromosome is defined as the list of variables used by a neural network for training and prediction. Therefore each chromosome defines a specific network's architecture, as described above. For example if a chromosome codes the following set of variables:

closing value of DAX, average change in 20 days, MACD oscillator value, percentage change in 20 days, then the network defined by this chromosome has 4 input neurons, 2 neurons in a hidden layer and 1 output neuron. For each input neuron the value of the respective variable is provided. The fitness function is reverse-proportional to the error of a network after training. The smaller the error on validation samples, the higher the fitness of a chromosome. For each chromosome several neural networks with different initial weights, but the same architecture are considered. The fitness function can be calculated with respect to either the minimum or the average error of all networks assigned to the chromosome. If the average error is used the algorithm is less sensitive to accidental increase or decrease of the learning process effectiveness caused by the initial weights choice.

There are two main operators in the GA. A crossover is responsible for creating new chromosomes using information coded by both parents. Mutation brings random changes in chromosomes, which expand their capabilities of searching for the result in the whole solution space.

Two crossover methods are implemented in the proposed system. The first one is classical crossover with one cutting point. The cutting point can by selected randomly or chosen according to the best fitness of the resultant child chromosome (see fig. 2). The alternative crossover method, a combined crossover, depends on the common part of two parent chromosomes and random selection of the rest of variables. The scheme of the combined crossover is presented in fig. 3. This kind of crossover promotes variables that are repeated in the population.

In any case, two offspring are created as a result. Selection of chromosomes for crossover is performed by the rank method. The probability of crossover equals 1, however, a parent is exchanged with assigned child only if the latter is better fitted. Using the above described crossover does not alter population size. Sizes of individual chromosomes generally change towards the preferred/winning ones.

During mutation variables coded by the chromosome are exchanged with variables chosen randomly from the remaining subset of all available ones. A single mutation can

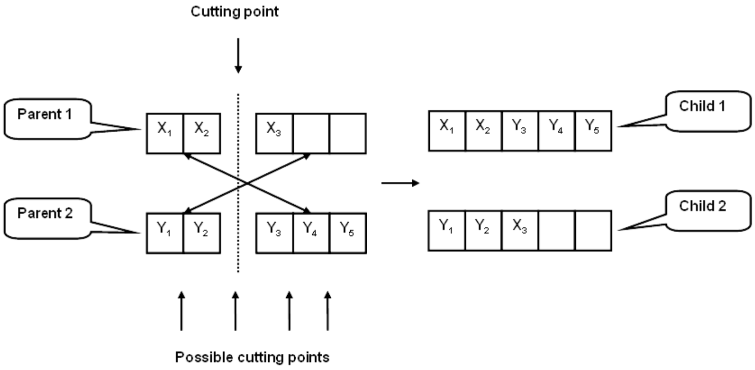


**Fig. 2.** The scheme of the classical crossover operation with one cutting point

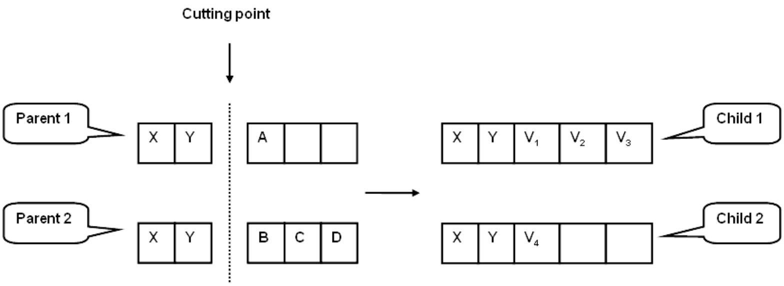


**Fig. 3.** The scheme of the combined crossover operation with a random selection of variables. $V_1, V_2, V_3, V_4$ are randomly selected from the remaining pool of variables (i.e. except for $X$ and $Y$) with additional condition that they are pairwise different within each of the chromosomes, i.e. $V_2 \neq V_3, V_3 \neq V_4$ and $V_2 \neq V_4$.

affect only one randomly selected variable. The probability of mutation depends on the type of the chromosome (dead or alive) - see section 4.

The first step of GA is creation of the initial population. Variables coded by chromosomes are randomly selected from the initial pool. Diversity of the population is assured by creating chromosomes of different sizes (numbers of coded variables). All chromosomes' sizes from a predefined range are represented in the initial population [12].

In each iteration of the GA new chromosomes are created as a result of crossover and mutation. Gradually the average error of prediction made by neural networks coded by the chromosomes decreases.

After every iteration the best chromosome is appointed. A chromosome can be considered the best one only if all neural networks used in the fitness evaluation are able to learn (is alive for every network). Finally, in the last iteration of the GA the network with the lowest error for the best chromosome is saved and subsequently used in the final test.

## 3 Summary of Results

In order to examine the effectiveness and usefulness of proposed system of prediction several simple prediction models were implemented for comparison. All of them are based on generating buy/sell signals to make the maximum profit of transactions on index value. In the initial state there is a fixed budget for transactions. At the end of the day the signal for the next day is generated. Transaction is done with the next day opening value. If the signal is "buy" as many indices as possible with the current budget are bought. Remaining money (the amount smaller than the cost of one index) becomes the budget for the next day. If the "sell" signal is provided all possessed indices are sold. The total budget is available for future transactions. The final result is the total budget after selling all indices at the last day.

**Model** 1 - *"buy and hold"* strategy: the first day's signal is *"buy"*, the last day's one is *"sell"*, and there are no other buy/sell signals generated. As a result the profit equals the change of an index value in a given time period.

**Model** 2 - signals are generated assuming that the direction of change of an index value of the next day will be the same as the last change direction.

**Model** 3 - signals are calculated using the next day prediction of a neural network - *this model*

**Model** 4 - signals are calculated using the MACD oscillator. Signals are generated at the crossing of lines of the oscillator [6].

**Model** 5 - signals are generated using the actual knowledge of the next day's value. This, prothetic and omnipotent, model gives the best possible result under defined conditions.

Models 1 and 2 are constructed using simple heuristics with no particular prediction system. Model 4 is based on the very popular MACD oscillator. Model 5 defines the upper limit of a profit that can be achieved assuming that the exact knowledge of future index values is available beforehand.

A single experiment is divided into steps. Each step consists of neural network training on 290 samples and 5 validation records. The number of training samples is chosen to cover a year of observations. On the other hand it is a reasonable number of samples for training the average-size neural network used in the proposed system. The neural network selected in the current step is tested on the subsequent 5 records not used in previous phases. In the next step the process is repeated using samples shifted by 5 records. Large part of the data is shared between steps since in subsequent ones the time-window is shifted by 5 days forward (test samples from immediately previous step become validation ones and validation samples become the last part of the training days).

Four different experiments (one for each model except for the buy and hold one), each consisting of 20 steps were performed in order to present usefulness of the above approach. Consequently, in each experiment the consistent set of 100 $(20 \times 5)$ samples was considered as the test data (covering the period from 2004/04/07 to 2004/08/26). In each step the following sequence of activities was performed: **A:** creating initial pool of variables, **B:** finding the best chromosome by the GA, and **C:** performing the final training with neural network and input variables coded by that chromosome.

The basic GA parameters were defined based on some preliminary tests. The population size was equal to 48, the number of generations was equal to chromosomes 500, inial sizes of the chromosomes were between 4 and 11, the number of neural networks used in each chromosome for fitness calculation was set to 3 and the number of iterations during neural network learning for fitness calculation in GA was equal to 200.

Results of experiments performed according to the above described scenarios are summarized in Table 1. In each experiment the amount of initial money was equal to $100,000$ units. Please recall, that Model 5 assumes the exact knowledge about the future. The result of Model 1 (which implements *buy and hold* strategy) is proportional to the change of index value in the considered period of time. Furthermore the repeatability of results was tested by running several experiments in the same time period and under the same experiment's settings except for initial neural networks' weights. The results are presented in Table 2. Moreover, a detailed analysis of the sets of variables chosen is subsequent steps of the experiments was performed, proving their "visible sense".

**Table 1.** Return of proposed models attained in different experiments

| **Model/Experiment** | **Return [%]** |
|---|---|
| Model 1 | -5.46 |
| Model 2 | -7.54 |
| Model 3 | 9.31 |
| Model 4 | -3.59 |
| Model 5 | 39.56 |

**Table 2.** Return [%] of proposed model attained in six experiments performed with the same algorithm settings and for the same period of time

| #1 | #2 | #3 | #4 | #5 | #6 | **avg.** |
|---|---|---|---|---|---|---|
| 9.31 | 4.68 | 3.51 | 3.41 | 5.96 | 7.67 | 5.44 |

The results confirm the hypothesis about changing in time dependencies between variables. The frequency analysis of choosing variables from three different stock markets to the current input set proved the hypothesis that stock markets are related and that the influence of NYSE on the GSE (generally, on European stock markets) is prevailing the one of Japanese TSE. These variables were also more preferable than the exchange rates. The analysis of the algorithm's variable selection process is presented in [12] and [13].

## 4 *Dead* and *alive* Chromosomes

One of the interesting conclusions drawn from the analysis of experimental data was the existence of specific sets of input variables for which the neural networks were unable to achieve acceptable level of an error. This usually happened in the initial population of chromosomes, but sometimes also in further generations, when some sets of the variables were not suitable for learning. Chromosomes which coded such useless sets were called *dead*. The remaining chromosomes were called *alive*. More precisely, a chromosome was defined as dead if and only if none of the three neural networks defining chromosome's fitness was unable to learn, i.e. achieved unsatisfactory test result on validation records (not used in the training).

Since chromosomes in the initial population are selected randomly (within some predefined ranges) the number of dead individuals at the beginning of the GA process can be relatively high. Despite the initial number of them, the system tends to successfully eliminate such useless chromosomes.

Two typical situations are presented in fig. 4, where two populations from two different steps of the same experiment are described. Each population consists of $48$ individuals. All algorithm's parameters are the same for both steps, except for different 5-day time periods covered by each step. In the population $A$ the number of dead chromosomes in the initial generation is relatively small and this situation does not drastically change until the final generation (though some fluctuations in the initial phase can be observed). The initial situation in population $B$ looks quite differently. Here the number of dead chromosomes is high (actually only some of them are alive), but the ratio

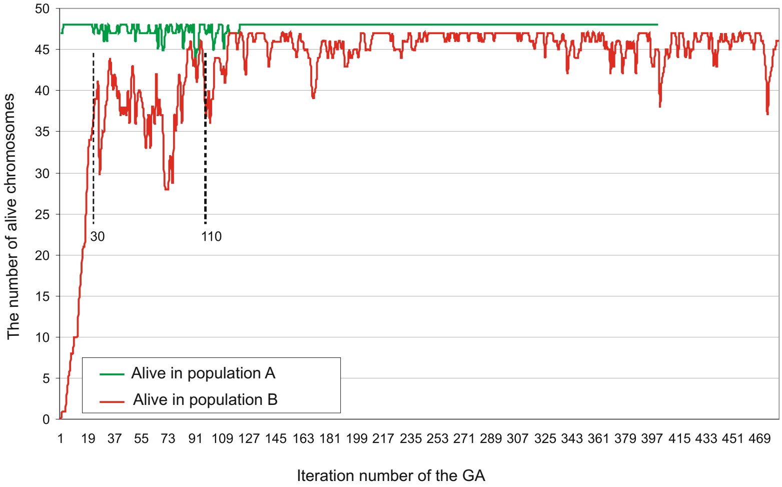


**Fig. 4.** The numbers of alive chromosomes in two exemplar populations. See description within the text

between alive and dead individuals is quite rapidly improving in the initial period of GA. After 30 generations this ratio becomes relatively high and from the 110th generation the number of alive chromosomes in the population stabilizes at the level of about 95%.

In order to alleviate the negative influence of dead chromosomes additional constraints were imposed on the mutation operator, whose probability depended on the type of a chromosome. Generally speaking, probability for dead chromosomes was much higher than for alive ones. This setting forced the undesirable chromosomes to be changed. Changes in alive chromosomes were rare - just enough to keep the diversity of the population. In particular, the probability of mutation of a dead chromosome was set to 1 in the whole experiment. The respective probability of alive chromosomes was higher in the initial phase of the GA (equal to 0.2) and lowered subsequently to 0.05. Furthermore, the probability of mutation of alive chromosomes was automatically increased (from 0.05 to 0.2) if the difference between the maximum fitness in the population and the average fitness was below some predefined threshold. Additionally, mutations of alive chromosomes were only allowed if the number of such chromosomes exceeded 90% or was less than 50% of the population size. In the remaining interval ($[50\%, 90\%]$) only dead chromosomes were allowed to be mutated.

The above steering parameters were crucial for proper maintenance of the population, especially for keeping the number of alive individuals at a high level. On the other hand some amount of dead chromosomes was also allowed to exist in the population in order to keep it divers and having potential for further development.

## 5 Conclusions and Future Work

A hybrid neuro-genetic method of prediction with application to financial task, which is prediction of the closing value of DAX - the German Stock Exchange index, is presented and examined in the paper. Assuming the flow of information between different stock markets and their mutual relations the proposed system uses data from five different sources: three stock markets and two exchange rates.

Due to changing dependencies between variables describing financial markets (indices, oscillators, exchange rates, etc.) and assuming that the usefulness of any variable is limited in time, the proposed approach relies on applying the GA for *frequent* input data selection (among the predefined pool of variables from the above mentioned five sources).

The standard GA procedure is enhanced by adding a new type of crossover operator and a mechanism that controls the range of chromosomes' sizes. Furthermore, the chromosomes are categorized as being either *alive* (i.e. the neural net architectures coded by them are able to learn) or *dead* (in the opposite case). Based on the ratio between alive and dead chromosomes the probabilities of genetic operators are changing accordingly in order to get rid of useless, non-alive chromosomes.

In future work we plan to apply a systematic procedure for selection system's internal parameters (population size, the ranges within which the dead chromosomes are allowed to be mutated, etc.). In the current version of the algorithm the above choices are the result of non-exhaustive trial and error procedure. Also more numerical evaluations are planned in order to further validate the efficacy of proposed neuro-evolutionary method.

## References

1. Thawornwong, S., Enke, D.: The adaptive selection of financial and economic variables for use with artificial neural networks. Neurocomputing 56, 205–232 (2004)
2. Fieldsend, J., Singh, S.: Pareto evolutionary neural networks. IEEE Transactions on Neural Networks 16(2), 338–354 (2005)
3. Dempster, M., Payne, T., Romahi, Y., Thompson, G.: Computational learning techniques for intraday FX trading using popular technical indicators. IEEE Transactions on Neural Networks 12(4), 744–754 (2001)
4. John, G., Miller, P., Kerber, R.: Stock selection using rule induction. IEEE Expert, 52–58 (1996)
5. Duch, W., Setiono, R., Zurada, J.: Computational intelligence methods for rule-based data understanding. Proceedings of the IEEE 92, 771–805 (2004)
6. Murphy, J.: Technical Analysis of the Financial Markets. New York Instituite of Finance (1999)
7. Cover, T.: An algorithm for maximizing expected log investment return. IEEE Transactions on Information Theory 30(2), 369–373 (1984)
8. Schwager, J.D.: Getting started in Technical Analysis. John Wiley & Sons, Chichester (1999)
9. Fahlman, S.E., Lebiere, C.: The cascade-correlation learning architecture. In: Touretzky, D. (ed.) Advances in Nueral Information Processing Systems, San Mateo, CA, USA, vol. 2, pp. 524–532. Morgan Kaufmann, San Francisco (1990)

10. LeCun, Y., Denker, J., Solla, S.: Optimal brain damage. In: Touretzky, D. (ed.) Advances in Nueral Information Processing Systems, vol. 2, pp. 598–605. Morgan Kaufmann, San Francisco (1990)
11. Hassibi, B., Stork, D.: Secong order derivatives for network pruning: Optimal Brain Surgeon. In: Touretzky, D. (ed.) Advances in Neural Information Processing Systems, San Mateo, CA, USA, vol. 5, pp. 164–171. Morgan Kaufmann, San Francisco (1993)
12. Mańdziuk, J., Jaruszewicz, M.: Neuro-evolutionary approach to stock market prediction. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2007), Orlanodo, FL, USA, pp. 2515–2520. IEEE Press, Los Alamitos (2007)
13. Jaruszewicz, M., Mańdziuk, J.: Neuro-genetic system for DAX index prediction. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 42–49. Springer, Heidelberg (2006)

# String Pattern Recognition Using Evolving Spiking Neural Networks and Quantum Inspired Particle Swarm Optimization

Haza Nuzly Abdull Hamed[1], Nikola Kasabov[1], Zbynek Michlovský[2],
and Siti Mariyam Shamsuddin[3]

[1] Knowledge Engineering and Discovery Research Institute (KEDRI),
Auckland University of Technology, New Zealand
{hnuzly,nkasabov}@aut.ac.nz
[2] Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
imichlov@fit.vutbr.cz
[3] Soft Computing Research Group, Universiti Teknologi Malaysia, Malaysia
mariyam@utm.my

**Abstract.** This paper proposes a novel method for string pattern recognition using an Evolving Spiking Neural Network (ESNN) with Quantum-inspired Particle Swarm Optimization (QiPSO). This study reveals an interesting concept of QiPSO by representing information as binary structures. The mechanism optimizes the ESNN parameters and relevant features using the wrapper approach simultaneously. The N-gram kernel is used to map Reuters string datasets into high dimensional feature matrix which acts as an input to the proposed method. The results show promising string classification results as well as satisfactory QiPSO performance in obtaining the best combination of ESNN parameters and in identifying the most relevant features.

**Keywords:** String Kernels, Text Classification, Evolving Spiking Neural Network, Particle Swarm, Quantum Computing.

## 1 Introduction

String pattern recognition is an approach for determining which group a string belongs to, according to its contents. This task, despite being quite challenging, is very important to certain areas such as internet security and virus detection. Strings can be texts, musical symbols or others which are not necessarily in numerical formats. Since most classifier algorithms can only accept numerical values, transformation from strings to numerical values is required. String kernels are a well-known method to transform string input values into high dimensional input vectors. There are several well-known string kernels such as Bag of Words (BOW) and N-gram Kernels. Output from the kernel process which is the kernel matrix is used as an input to the algorithm for classification, clustering or ranking tasks. This technique is quite simple yet quite effective to transform input from string to the numerical value.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 611–619, 2009.

The third generation neural network simulates how biological neurons send and receive information based on spikes. This neural simulation increases the level of realism where the neuron is only fired when it reaches a specific value of electrical charge. Several improvements and variants of the spiking neuron model have been developed. According to Kasabov [1], the Evolving Spiking Neural Network (ESNN) as proposed by Wysoski *et al.* [2] states that the output neuron evolves based on weight vector similarities. In this study, the ESNN model is selected as the string classifier. Like other neural network models, the correct combination of ESNN parameters could influence the classification performance of the network. Apart from parameter value, the more features used, does not necessarily result in better or higher levels of classification accuracy. In some cases, having fewer significant features might help the network to reduce the processing time and produce good classifications. Therefore, feature optimization could be considered as an important pre-processing tool in a classification task. Due to this reason, Quantum-inspired Particle Swarm Optimization (QiPSO) is proposed in this study as the new optimizer model to investigate its efficiency in optimizing ESNN parameters as well as selecting the most relevant features from the string kernel matrix which both have a direct influence to the classification accuracy.

This paper is organized as followed; Section 2 briefly explains the ESNN model and Section 3 discusses the chosen optimizer model; the QiPSO and how it is derived from standard PSO. Section 4 deals with the introduction to string kernels. The framework of the proposed method is presented in Section 5. This section explains how the features are selected from the string matrix and ESNN parameters are optimized using QiPSO. Section 6 provides the details of the experiment setup and results, followed by conclusion of this study with details on future work in Section 7.

## 2 Evolving Spiking Neural Network

Hopfield [3] presented a model of spiking neurons in 1995. Since then, there have been several enhancements and variants of the spiking neuron model. The architecture of ESNN in this paper is based on Wysoski's model [2]. This model consists of the encoding method of real value data to spike time, network model and learning method.

There are several information encoding methods in Spiking Neural Network (SNN). However, our implementation is based on Population Encoding as proposed in [4] where a single input value is encoded to multiple input neurons of *M*. Each input neuron represents a certain spikes of firing time. The firing time of a input neuron *i* is calculated using the intersection of Gaussian function. The centre is calculated using Equation 1 and the width is computed using Equation 2 with the variable interval of [$I_{\min}$, $I_{\max}$]. The parameter $\beta$ controls the width of each Gaussian receptive field.

$$\mu = I_{\min} + (2*i-3)/2*(I_{\max} - I_{\min})/(M-2)\,. \tag{1}$$

$$\sigma = 1/\beta(I_{\max} - I_{\min})/(M-2) \quad where \quad 1 \le \beta \le 2\,. \tag{2}$$

Thorpe [5] proposed a simplified spiking neuron model referred to as the Fast Integrate and Fire Model. The fundamental aspect of this model is that the earliest spikes received by a neuron will have a stronger weight compared to the later spikes. Once the neuron reaches a certain amount of spikes and the Post-Synaptic Potential (PSP) exceeds the threshold value, it will fire and becomes disabled. The neuron in this model can only fire once. The computation of PSP of neuron *i* is presented in Equation 3,

$$u_i = \begin{cases} 0 & if \;\; fired \\ \sum w_{ji} * Mod_i^{order(j)} & else \end{cases} \quad . \tag{3}$$

where $w_{ji}$ is the weight of pre-synaptic neuron *j*; $Mod_i$ is a parameter called modulation factor with interval of [0,1] and *order(j)* represents the rank of spike emitted by the neuron. The *order(j)* starts with 0 if it spikes first among all pre-synaptic neuron and increases according to the firing time.

In the One-pass Learning algorithm, each training sample creates a new output neuron. Trained threshold value and the weight pattern for the particular sample are stored in the neuron repository. However, if the weight pattern of the trained neuron is considered too similar with the neuron in the repository, the neuron will merge into the most similar one. The merging process involves modifying the weight pattern and the threshold to the average value. Otherwise, it will be added as a newly trained neuron to the repository. The major advantage of ESNN is the ability of the trained network to produce new samples without retraining. The detailed ESNN learning algorithm as described in [6] is depicted below. More details on ESNN can be found in [2], [6] and [7].

**Algorithm 1.** ESNN Training Algorithm

```
Encode input sample into firing time of pre-synaptic neuron $j$
Initialize neuron repository $R$ ={ }
Set ESNN parameters $Mod$ = [0,1], $C$ = [0,1] and $Sim$ = [0,1]
for all input sample $i$ belong to the same output class do
    Set weight for all pre-synaptic neuron where: $w_j = (Mod)^{order(j)}$
    Calculate $PSP_{\max(i)} = \sum w_j * Mod^{order(j)}$
    Get PSP threshold value $\theta_i = PSP_{\max(i)} * C$
    if the trained weight vector <= $Sim$ of trained weight in $R$ then
        Merge weight and threshold value with most similar neuron
        $w = \frac{w_{new} + w * N}{N + 1}$
        $\theta = \frac{\theta_{new} + \theta * N}{N + 1}$
        where $N$ is number of merge before
    else
        Add new neuron to output neuron repository $R$
    end if
end for (Repeat  to all input samples for other output class)
```

## 3 Quantum Inspired Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population based optimization technique developed by Eberhart and Kennedy in 1995 [8]. Individuals in PSO work together to solve a given problem by responding to their own performance and the performance of the other particles in the swarm. Each particle has their own fitness value calculated during the optimization process and the best fitness value achieved so far is stored and normally referred to as personal best or individual best (*pbest*). The overall best fitness value obtained by any particle in the population so far is called global best (*gbest*) which stores the solution. Every particle is accelerated towards a new position by calculating the position velocity where the value of *pbest* and *gbest* would influence the direction of the particle in the next iteration. Equation 4 illustrates the velocity update and Equation 5 is the calculation of the new particle position.

$$v_n = w * v_{n_{t-1}} + c_1 * rand() * (g_{best_n} - x_n) + c_2 * rand() * (p_{best_n} - x_n) . \tag{4}$$

$$x_n = x_{n_{t-1}} + v_n . \tag{5}$$

Value of the random number is between 0 and 1. $C_1$ and $C_2$ control the particle acceleration towards personal best or global best.

Han and Kim proposed a Quantum Evolutionary Algorithm (QEA) in 2002 [9] which was inspired by the concept of quantum computing. According to the classical computing concept, information is represented in bits where each bit must hold either 0 or 1. However, in quantum computing, information is instead represented by a qubit in which a value of a single qubit could be 0, 1, or a superposition of both. Superposition allows the possible states to represent both 0 and 1 simultaneously based on its probability. The quantum state is modeled by the Hilbert functions and is defined as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ where $\alpha$ and $\beta$ are complex numbers defining probabilities at which the corresponding state is likely to appear (when a qubit collapses, for instance, when reading or measuring). Probability fundamentals require that $|\alpha|^2 + |\beta|^2 = 1$, where $|\alpha|^2$ gives the probability that a qubit is in the OFF (0) state and $|\beta|^2$ gives the probability that a qubit is in the ON (1) state. General notation for an individual with several qubits can be defined as $\begin{bmatrix} \alpha_1 & | \; \alpha_2 & | \; ..... & | \; \alpha_m \\ \beta_1 & \beta_2 & ..... & \beta_m \end{bmatrix}$.

QiPSO was first discussed in [10] and there are several variants of QiPSO. The main idea of QiPSO is to use a standard PSO function to update the particle position represented in a qubit. In order for PSO to update the probability of a qubit, the quantum angle, $\theta$, is used. Quantum angle $\theta$ can be represented as $\begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$, and it is equivalent to $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ which satisfies the probability fundamental of $|\sin(\theta)|^2 + |\cos(\theta)|^2 = 1$. Therefore, the qubits value can be replaced by $[\theta_1 \mid \theta_2 | ..... | \theta_m]$ in QiPSO. The velocity update formula in standard PSO is modified to get a new quantum angle which is translated to the new probability of the qubit by using Equation 6.

$$\Delta\theta_n = w * \Delta\theta_{n_{t-1}} + c_1 * rand() * (\theta_{gbest_n} - \theta_n) + c_2 * rand() * (\theta_{pbest_n} - \theta_n) \cdot \tag{6}$$

Then, based on the new $\theta$ velocity, the new probability of $\alpha$ and $\beta$ is calculated using a rotation gate as follows:

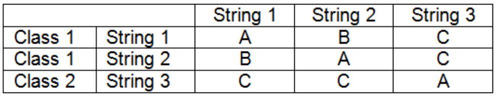
$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_{t-1} \\ \beta_{t-1} \end{bmatrix} . \tag{7}$$

or simply by replacing the rotation gate with $\theta = \theta_{t-1} + \Delta\theta$ where $\theta$ is the new quantum angle of the quantum particle position.

## 4 String Kernels

In order to allow our proposed method to operate on string data, string kernels were used to transform the input data to the desired input format. Kernel provides the classifier algorithm with the capability of mapping the original non-linear separable data into a higher-dimensional space which is linearly separable. This method is normally referred as the Kernel Trick [11].

The *n*-grams approach has been chosen where *n*-grams are *n* adjacent characters (substring) from the alphabet *A* [12]. For example, if *n*=3 for the string “KEDRI”, the trigrams output will be {KED, EDR, DRI}. Based on this approach, the similarity between strings is calculated and comes out with the kernel matrix. This process is illustrated in Fig. 1.



| | | String 1 | String 2 | String 3 |
|---|---|---|---|---|
| Class 1 | String 1 | A | B | C |
| Class 1 | String 2 | B | A | C |
| Class 2 | String 3 | C | C | A |

**Fig. 1.** Kernel matrix from three strings. String 1 and 2 are class 1 and String 3 is class 2. Comparison between same strings will give the highest similarity value of A, the similarity calculation between the same classes will give value of B which is slightly lower than A while the similarity value between different classes is C which is the lowest. Based on this similarity value, the feature pattern between input samples can be produced.

## 5 A Proposed ESNN-QiPSO for String Pattern Recognition

ESNN is chosen as the classifier of the string data in this experiment. String data is translated into the input features using *n*-gram kernel. In order to produce a better classification accuracy, all three ESNN parameters namely Modulation Factor (*Mod*), Proportion Factor (*C*) and Neuron Similarity (*Sim*) as well as input features are optimized using QiPSO. Optimization of the ESNN using QEA was addressed by Schliebs *et al.* [6] recently. However, this paper introduces QiPSO as a new optimizer for ESNN. From the well known wrapper approach, QiPSO interacts with an induction method, and in this case, ESNN; produces the best set of the ESNN parameters and identifies relevant features. All particles are initialized with a random set of binary values and from there they interact with each other based on classification accuracy. Since there are two components to be optimized, each particle is divided into two parts. The first part of each particle holds the feature mask value. This value is used for the feature optimization while the other part holds a binary string for parameter optimization.

For the parameter optimization, a set of qubits represents the parameter values. Since information held by a particle is in a binary representation, a conversion into a real value is required. For this task, the Gray code method is chosen since it is proven to be a simple and effective way to represent real values with a binary representation. In feature optimization, value 1 indicates the feature is selected, and value 0, otherwise. Input features are taken from the kernel matrix. Fig. 2 illustrates the proposed design of ESNN-QiPSO for string classification.

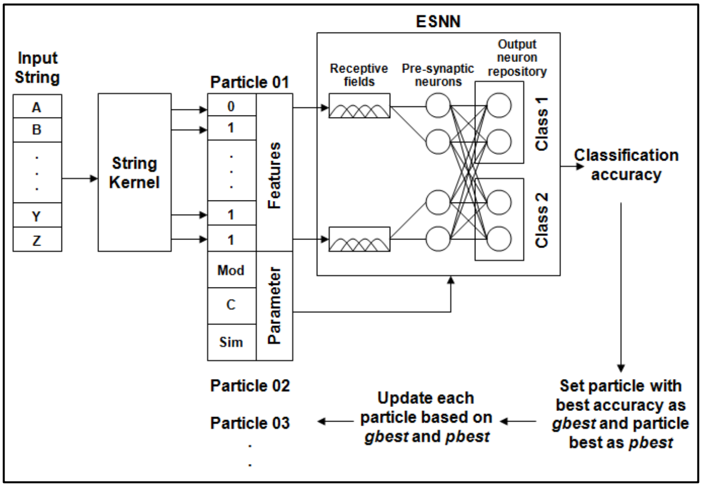


**Fig. 2.** A proposed ESNN-QiPSO for string classification

## 6 Experimental Setup and Results

We have used the labeled and reformatted Reuters 21578 string dataset from UCI Repository [13]. Only relevant information from tags topic, title and body text were extracted and some unknown tags such as "&", "$" were removed. Finally, all characters were changed to lowercase. The problem consisted of 150 samples with 38 from acquisition, 36 from corn, 38 from crude and 38 from earn. Parameters chosen were n=4 and λ = 0.5 in n-gram kernels where λ = [0, 1] represents the number of occurrences. 3-fold cross validation was used in this experiment.

Receptive fields were used to produce a weight pattern or weight vector of a particular sample that can identify the output class. Different numbers of receptive fields for each dataset influenced the accuracy of the result. From our preliminary experiment, 10 receptive fields were chosen. 20 particles have been used to explore the solution.

Number of dimension refers to the number of variables to be optimized by QiPSO which are three ESNN parameters and 150 features. Since all three parameters ranged between 0 and 1, six qubits were sufficient to represent the real value. $C_1$ and $C_2$ were set to 0.05 which meant a balanced exploration between *gbest* and *pbest* as well as the inertia weight $w = 1.5$. The dataset has been applied to ESNN with feature optimization and also ESNN with all features, both algorithms with parameter optimization. Because of the computation complexity, we run the experiment for six continuous times and the average result was computed in 300 iterations for both algorithms.

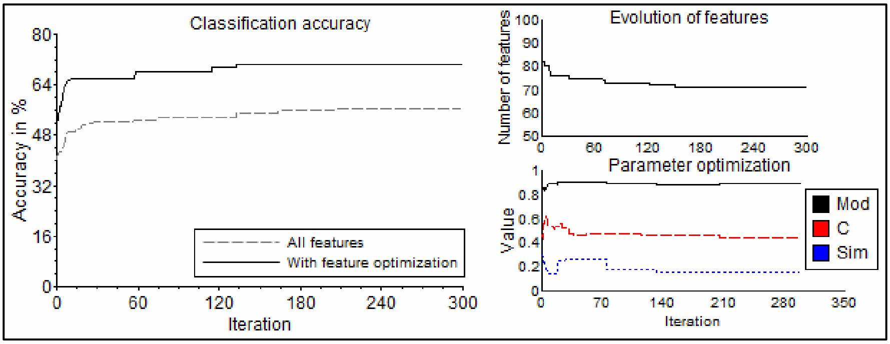


**Fig. 3.** Average classification result and parameter optimization

Fig. 3 shows the evolution of classification accuracy. The average accuracy for the ESNN with feature optimization is above 70% compared to ESNN using all features with the average of 55%. The poor accuracy of ESNN with all features is due to several input features from the kernel matrix containing information which cannot be used to differentiate output classes. These irrelevant features act as a noise which leads to a low classification accuracy. However, ESNN with feature optimization is able to reduce these irrelevant features; hence higher classification accuracy is obtained. From the total number of 150 input features, QiPSO is capable of reducing the features up to 70 features in 300 iterations. We strongly believe that these 70 features are the significant features because of its capability to produce higher accuracy. Since the 20 particles started the evaluation process by picking random features, we found that the average number of initial features selected by the *gbest* particle is around 80 features. The *gbest* particle always keeps best information and the best accuracy. Other particles update their position according to *gbest* as well as *pbest* until the new best particle is met. This procedure is repeated to eliminate irrelevant features for better identification of the most relevant features.

The similar procedure was also used to find the best combination of ESNN parameters. In this study, QiPSO manages to optimize binary string information which represents the ESNN parameter values. These parameters are in pair and are very closely related to each other. Overall, all three parameters evolve steadily towards a certain optimal value, and with correct combination can lead to better classification accuracy.

## 7 Conclusion and Future Work

This paper presents a novel method for string classification using ESNN-QiPSO. The results have shown that ESNN with parameter optimization and with using a small number of features produces promising results that is significant for future exploration. We have done some preliminary investigation to feed the string kernel matrix into Backpropagation Multi Layer Perceptron. Given the learning rate is set to 0.1, momentum rate is 0.9 with 120 hidden neurons, the training classification accuracy is around 55% in 300 iterations with testing result is around 50%. Since this experiment is in a very early stage and more experiments need to be conducted, we are planning to publish the result in the future. For future work, we are also going to apply our proposed method to real data from National Institute of Information and Communications Technology (NiCT), Japan. Other work includes how to find a more effective method for choosing the most relevant features and eliminating irrelevant features. Recurrent ESNN and integrative Probabilistic Spiking Neural Network as proposed in [14] will also be explored for string classification.

## Acknowledgments

We are grateful to the National Institute of Information and Communications Technology (NiCT) Tokyo, Japan for providing us the data, problem specification and support.

## References

1. Kasabov, N.: Evolving Connectionist Systems: The System Engineering Approach, 2nd edn. Springer, New York (2007)
2. Wysoski, S.G., Benuskova, L., Kasabov, N.: On-Line Learning with Structural Adaptation in a Network of Spiking Neurons for Visual Pattern Recognition. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4131, pp. 61–70. Springer, Heidelberg (2006)
3. Hopfield, J.: Pattern Recognition Computation Using Action Potential Timing for Stimulus Representation. Nature 376, 33–36 (1995)
4. Bohte, S.M., Kok, J.N., La Poutre, H.: Error-Backpropagation in Temporally Encoded Networks of Spiking Neurons. Neurocomputing 48(1) (2002)
5. Thorpe, S.J.: How Can The Human Visual System Process A Natural Scene in Under 150ms? Experiments and Neural Network Models. In: ESANN (1997)
6. Schliebs, S., Defoin-Platel, M., Kasabov, N.: Integrated Feature and Parameter Optimization for an Evolving Spiking Neural Network. In: Köppen, M., et al. (eds.) ICONIP 2008, Part I. LNCS, vol. 5506, pp. 1229–1236. Springer, Heidelberg (2009)
7. Schliebs, S., Defoin-Platel, M., Worner, S., Kasabov, N.: Integrated Feature and Parameter Optimization for an Evolving Spiking Neural Network: Exploring Heterogeneous Probabilistic Models. Neural Networks 22, 623–632 (2009)
8. Eberhart, R., Kennedy, J.: A New Optimizer Using Particle Swarm Theory. In: Proc. Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39–43. IEEE Press, NJ (1995)

9. Han, K.H., Kim, J.H.: Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization. IEEE Transactions on Evolutionary Computation 6, 580–593 (2002)
10. Sun, J., Feng, B., Xu, W.: Particle Swarm Optimization with Particles Having Quantum Behavior. In: Proc. Cong. Evolutionary Computation, CEC 2004, vol. 1, pp. 325–331 (2004)
11. Aizerman, M., Braverman, E., Rozonoer, L.: Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning. Automation and Remote Control 25, 821–837 (1964)
12. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text Classification Using String Kernels. Journal of Machine Learning Research 2, 419–444 (2002)
13. UCI Machine Learning Repository, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
14. Kasabov, N.: Integrative Probabilistic Evolving Spiking Neural Networks Utilising Quantum Inspired Evolutionary Algorithm: A Computational Framework. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008, Part I. LNCS, vol. 5506, pp. 3–13. Springer, Heidelberg (2009)

# Fault Condition Recognition Based on PSO and KPCA

Hongxia Pan, Xiuye Wei , and Xin Xu

School of Mechanical Engineering and Automation,
North University of China, Xueyuan Road 3,
030051 Taiyuan, China
panhx1015@163.com

**Abstract.** A method of kernel principal component analysis (KPCA) based on particle swarm optimization (PSO) is presented, which is applied in fault condition recognition of gear box. Comprehensively considered within-class scatter and between-class scatter of samples feature, the fitness function of kernel function parameter optimized is constructed, and the particle swarm optimization algorithm with adaptive accelerate (CPSO) is applied to optimize it. This method is applied to gear box condition recognition, compared with the recognized results based on principal component analysis (PCA). The results show that KPCA optimized by CPSO can effectively recognize fault conditions of gear box by reducing bind set-up of kernel function parameter, and its results of fault recognition outperform those of PCA. The conclusion is that KPCA based on PSO has advantage in nonlinear feature extraction of mechanical failure, and is helpful for fault conditional recognition of the complicated machine.

**Keywords:** particle swarm optimization (PSO), kernel principal component analysis (KPCA), fault diagnosis, gearbox.

## 1 Introduction

When faults appear in mechanical equipment, the fault information is weak and is usually accompanied by the occurrence of non-linear behavior, and it is difficult to differentiate different kinds of fault modes, so it is not helpful for identification and diagnosis of the fault condition. Kernel principal component analysis (KPCA) is a non-linear method, which is put forward by Schoelkopf and so on in the study of support vector classification algorithm [1]. It achieves some kind of non-linear mapping by the help of kernel function and maps the input vector to the high-dimensional feature space through non-linear mapping, enabling them to have a better divisibility in the high-dimensional feature space. Then analyze the principal element of the mapping data of high-dimensional space and obtain the non-linear principal component of raw data. KPCA takes full advantage of the idea of nuclear and therefore is more suitable to solve nonlinear problems. Currently, Lee Jong-min [2], Liao Guanglan [3], He Qingbo [4] and so on have applied KPCA to condition recognition and fault diagnosis and have obtained good results. However, for a given data set, the classification accuracy of KPCA algorithm is affected by the kernel function parameters, therefore, selecting the best kernel function parameter (shortly for kernel parameter) is critical.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 620–629, 2009.

At present, a large number of tests are mainly made manually or adopt cross-examination method, not only inefficient and time-consuming and laborious, but also the kernel function parameter determined this way is not necessarily optimal. Therefore, the study of kernel parameter optimization algorithm has a great significance on the establishment fault classifier and its performance optimization.

Particle Swarm Optimization (PSO) method is a population-based global parallel optimization method, in recent years, it has been widely used in function optimization, automatic control, machine learning, artificial life and so on other fields. This paper tries to use a method that combines Fisher discriminant function and particle swarm algorithm to solve the optimization problem of kernel parameter. First, establish the kernel parameter optimization model by using the idea of Fisher discriminant function, then use the improved particle swarm optimization algorithm to derive the global optimal solution of the parameter optimization model. Apply the optimized KPCA method into gear box fault condition recognition and compare it with recognition result of principal component analysis.

## 2 PSO Algorithm with Adaptive Accelerate

Usually for the population-based optimal method, early in the optimization period particles should be encouraged to move throughout the search space rather than gathering around the local maximum. On the other hand, late in the optimization period, increasing the convergence rate of the trend optimal solution to efficiently find out the optimal solution is very important. Taking these relevant factors into account, on the basis of standard particle swarm optimization algorithm, we have proposed the dynamic accelerate constant as a new parameter adaptive strategy, the following calls particle swarm optimization algorithm of dynamic accelerate constant as particle swarm optimization algorithm with adaptive accelerate (CPSO). Improved algorithm is aimed at encouraging particles to move throughout the entire search space in the early period of optimization and in the late period increasing the convergence rate of the trend convergence rate [5].

This improvement in the standard PSO algorithm, namely type (1) and (2), has realized accelerate constant $c_1$ and $c_2$ to change linearly along with the evolution algebra. The mathematical expressions are shown in type (3) and (4)

$$v_{id}(t+1)=\omega v_{id}(t)+c_1 r_1(p_{id}-x_{id}(t))+c_2 r_2(p_{gd}-x_{id}(t)) \tag{1}$$

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \tag{2}$$

Of which,

$$c_1=R_1+R_2*t/T_{max} \tag{3}$$

$$c_2=R_3+R_4*t/T_{max} \tag{4}$$

Of which, $R_1$, $R_2$ , $R_3$ and $R_4$ are fixed values set initially, $t$ and $T_{\max}$ are current iteration step and maximum iterative step respectively.

Test this algorithm in neural network, by simulation experiment and through a comprehensive comparison, it can be found that, $0 < R_1 + R_2 < 2$ should be set so that the iterative curve has a characteristic of faster convergence. The two values too large is not helpful for the development of particles, of which $R_1$ should not be too small, otherwise its single particle optimization effect can not be reflected in initial speed evolution iteration, so that $R_3 - R_4 > 2$ should be set and $R_3$ should be appropriately increased to enhance the space search capability to get the right seeds. Considering the range of speed difference $R_3 - R_4$ should not be too large, so that in the early period of search particle swarms have a better ability to explore new space, and the linear decreases as the iteration goes on, while the local search ability increases linearly ,and the optimal values can be obtained by increasing local development ability.

# 3 Kernel Parameter Optimization Based on Particle Swarm Algorithm

## 3.1 Principle of Kernel Principal Component Analysis

Kernel principal component analysis (KPCA) projects the input space to feature space through nonlinear mapping, and then makes principal component analysis on the mapping data of feature space to obtain the data projection on non-linear PCA. This nonlinear mapping is realized by inner product operation, namely it only needs to compute the kernel function which is used for inner product operation in original space, regardless of the specific realization form of the non-linear mapping. Thereby, it is called kernel function principal component analysis. By choosing different forms of kernel function, it can handle a large number of nonlinear problems [6]. For kernel methods, the kernel function which have obtained the most research are mainly polynomial kernel function, radial basis kernel function as well as neural network kernel function and so on. However the classification results obtained by choosing different kernel functions are extremely similar [7], in order to simplify the calculation, here we only choose the radial basis kernel function, the specific form of kernel function is as follows,

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{w}\right) \tag{5}$$

## 3.2 Additional Establishment of Kernel Function's Optimization Model

The main idea of kernel principal component transformation is mapping the training sample data to high-dimensional feature space, and then creating the optimal separating hyperplane in the feature space, so that the distance between the two types of data points and the hyperplane is the largest. However, in the traditional pattern recognition, the idea of Fisher discriminant function is just the opposite. The basic idea of Fisher discriminant function is finding an optimal projection direction, and projecting the raw data along the direction to get a straight line, so that projection data can be well separated, namely, distance between data of different models should be as large as possible and dispersion between data of the same model should be as small as possible. Therefore, we can draw on the idea of Fisher discriminant function to seek ways to optimize the kernel parameter [8].

The model establishment of particle swarm optimization kernel parameter based on fisher guideline is as follows [9].

Set $X_1(x_{11}, x_{12}, \cdots x_{1i}), X_2(x_{21}, x_{22}, \cdots x_{2j})$ are the two types of characteristic samples in feature space, of which $i = 1, 2, \cdots n_1,\ j = 1, 2, \cdots n_2$ 。

The mean vector of the two types in the feature space is,

$$\mu_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} \Phi(x_{1i}) \tag{6}$$

$$\mu_2 = \frac{1}{n_2} \sum_{j=1}^{n_2} \Phi(x_{2j}) \tag{7}$$

The square of between-type distance,

$$\begin{aligned} s_b &= \|\mu_1 - \mu_2\|^2 = (\mu_1 - \mu_2)^T (\mu_1 - \mu_2) \\ &= \frac{1}{n_1^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} k(x_{1i}, x_{1j}) - \frac{2}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} k(x_{1i}, x_{1j}) + \frac{1}{n_2^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} k(x_{2i}, x_{2j}) \end{aligned} \tag{8}$$

The square of dispersion within $X_k$ is,

$$\begin{aligned} s_{w1} &= \sum_{i=1}^{n_1} \|\Phi(x_{1i}) - \mu_1\|^2 \\ &= \sum_{i=1}^{n_1} \Phi(x_{1i})^T \Phi(x_{1i}) - n_1 \mu_1^T \mu_1 \\ &= \sum_{i=1}^{n_1} k(x_{1i}, x_{1i}) - \frac{1}{n_1} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} k(x_{1i}, x_{1j}) \end{aligned} \tag{9}$$

$$s_{w2} = \sum_{i=1}^{n_1} \|\Phi(x_{2i}) - \mu_2\|^2 = \sum_{j=1}^{n_2} k(x_{2j}, x_{2j}) - \frac{1}{n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} k(x_{2i}, x_{2j}) \tag{10}$$

Establish fitness function of PSO according to Fisher discriminant criteria.

$$F(w) = \frac{s_{w1} + s_{w2}}{s_b} \tag{11}$$

Seek the minimum value of fitness function to ensure that dissemination between-class the maximum and inner-class the minimum. Set $w^*$ the minimum value of Fisher discriminant function. A large number of experiments show that, for completely non-linear separable problem, the minimum value $w^*$ exists. For linearly separable or almost linearly separable problem, with $w$ changes from small to large, $F(w)$ declines dramatically, and then trends to stabilize, now we can choose $w$ when $F(w)$ trends to stabilize as $w^*$.

### 3.3 Steps of Kernel Parameter of PSO

1) Through equation (8), (9), (10), calculate the square of between-class distance $s_b$ and the square of inner-class dispersion $s_{w1}, s_{w2}$.

2) Through equation (8), Construct Fisher discriminant function $F(w)$, and take $F(w)$ as the adapting function of particle swarm optimization.

3) Set the range of $w$ $(w_{\min}, w_{\max})$, and swarm scale $m$ of CPSO, iteration number $T_{\max}$, constant learning factors $R_1, R_2, R_3, R_4$ ,the maximum and minimum inertia weight $\omega_{\max}, \omega_{\min}$ as well as the maximum limit speed $V_{\max}$.

4) Generate $w$ initial group randomly, and calculate individual adapting value $F_p$ and overall adapting value $F_g$ of the swarm.

5) Judge whether the iteration number reaches the termination condition. If $t < T\max$, repeat steps (4) and (5), otherwise output the current solution as the optimal solution $w^*$ and terminate the algorithm.

Kernel function parameter optimization based on PSO is shown in Fig. 1.

## 4 Gear Box Fault Condition Recognition Based on KPCA

In mechanical equipment, the gear box is usually used to change the rotational speed and transmit power, and it is vulnerable to damage and fault due to bad working conditions and other reasons; As annular gear of gear box、 bearings, shaft and so on are affected by complex factors such as the installation location, operating conditions and so on, gear box fault diagnosis is a very complex issue, and the relationship between its fault and sign is not very clear, which is a non-linear mapping relationship .Kernel principal component analysis can not only extract non-linear characteristics of gear box vibration signals, but also carry on the fault condition recognition, experiments show that this method is very effective for gear box fault detection and recognition.

### 4.1 Establish of Gearbox Feature Parameters Set

Take laboratory JZQ250 gear box as study object and conduct fault simulation experiment. On the basis of having measured the acceleration speed signal of six measurements of different rotational speed, conduct signal processing and extract 27 feature parameters as raw feature parameter set, of which 21 time-domain feature parameters, including mean, mean-square value, root-mean-square value, root amplitude, average, absolute amplitude, skewness, kurtosis, maximum value, minimum value, variance, kurtosis index, waveform, peak index, pulse index, margin target, kurtosis index, skewness index, skewness factor, eight-order moment factor, sixteent-order moment factor, six-order moment factor. The six frequency-domain feature parameters include, gravity indicator of power spectrum, mean-square spectrum, frequency-domain variance, relevant factor, harmonic factor, origin moment spectrum. Take 27 parameters as the original features in the study. Analyze data of 2 measurement point which locates at the right bearing seat of intermediate shaft and 5 measurement point which locates at the intermediate shaft on the left surface of the box, and extract 60 groups of samples respectively from the fault such as normal, input shaft bent, intermediate shaft float, ring fault inside and outside the bearing, tooth wear, tooth collapse and so on. Take 30 samples as the training group and 30 groups as the test samples, then carry on kernel principal component analysis and recognition of fault condition of gearbox.

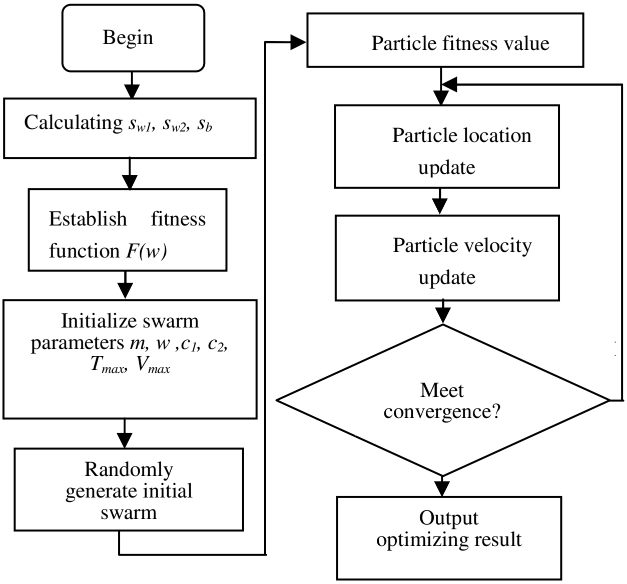

Begin
Calculating $s_{w1}$, $s_{w2}$, $s_b$
Establish fitness function $F(w)$
Initialize swarm parameters $m$, $w$ ,$c_1$, $c_2$, $T_{max}$, $V_{max}$
Randomly generate initial swarm
Particle fitness value
Particle location update
Particle velocity update
Meet convergence?
Output optimizing result

**Fig. 1.** Flow chart of kernel parameter optimization based on PSO

## 4.2 Optimization of Kernel Function Parameter Based on PSO

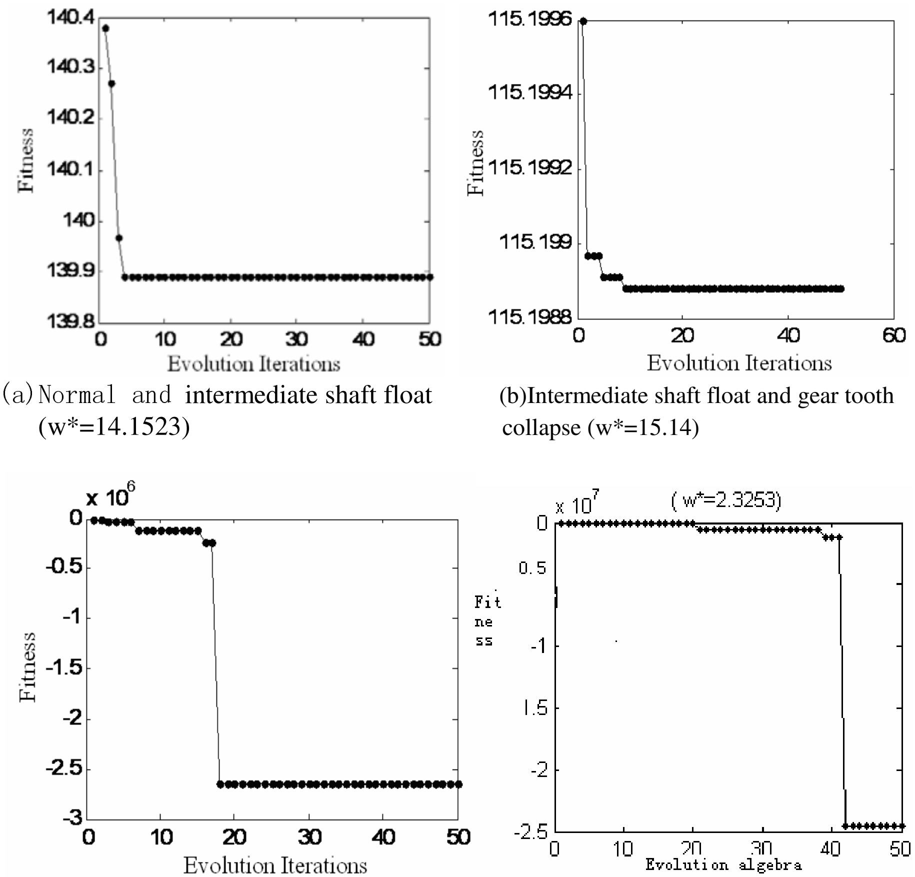
In kernel principal component analysis, radial basis kernel function width $w$ is the guarantee for good performance of kernel learning. $w$ too large, **spheres influence** of samples will be too large, so some training samples that have no relationship will

**Table 1.** Parameter set table of CPSO

| CPSO | $m$ | $\omega_{\max}$, | $\omega_{\min}$ | $R_1$ | $R_2$ | $R_3$ | $R_4$ | $T_{\max}$ | $V_{\max}$ |
|---|---|---|---|---|---|---|---|---|---|
| Parameter | 20 | 1.2 | 0.4 | 1 | 0.5 | 6 | 2 | 50 | 1 |

**Table 2.** KPCA result of samples under different fault condition

| | Kernel function width $w^*$ | Fitness $F(w)$ |
|---|---|---|
| Intermediate shaft float and gear tooth collapse | 14.52 | 139.8 |
| Normal and inner ring of bearing | 0.6176 | -2652801.9 |
| Noamal and intermediate shaft float | 15.74 | 115.1989 |
| Inner ring of bearing and tooth wear | 2.3253 | -24581918.1 |



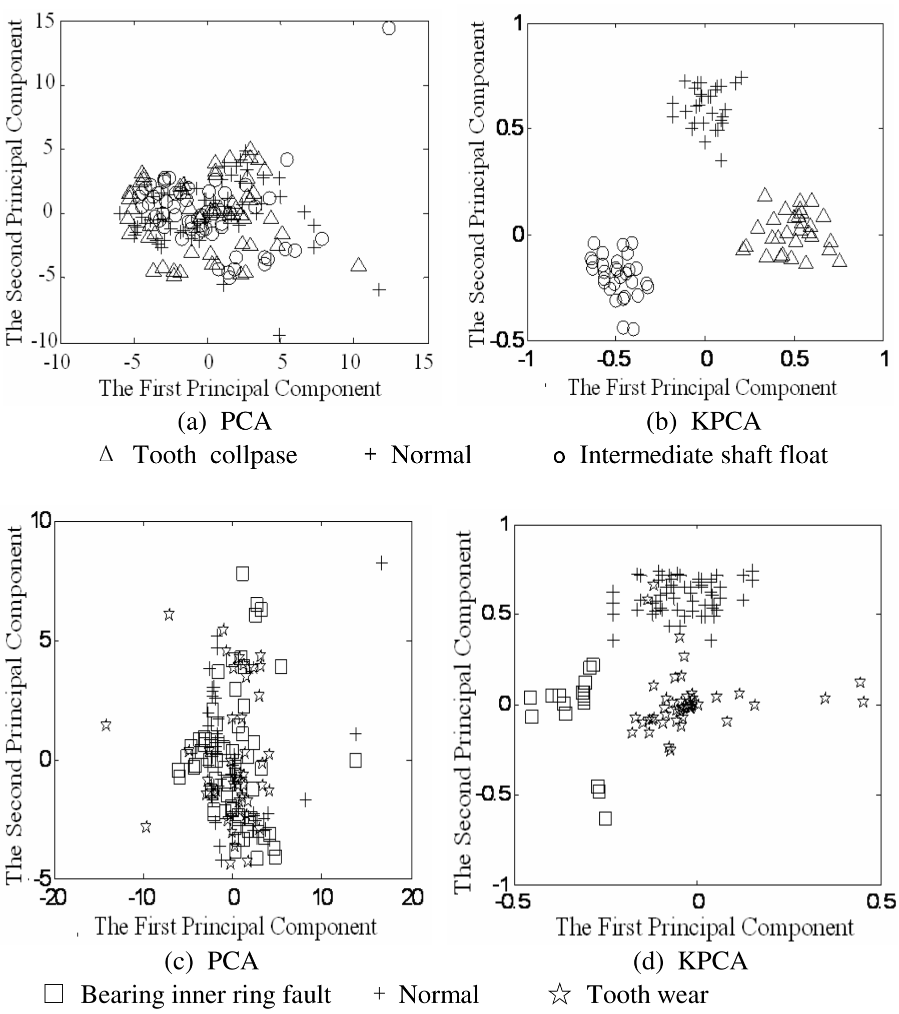
(a) Normal and intermediate shaft float (w*=14.1523)

(b)Intermediate shaft float and gear tooth collapse (w*=15.14)

(c) Normal and inner ring of bearing (w*=0.6176)

(d) Inner ring of bearing and tooth wear

**Fig. 2.** Evolution process of adapting function based on CPSO



(a) PCA

(b) KPCA

(c) PCA

(d) KPCA

**Fig. 3.** First and second projection of PCA and KPCA of different faults

interfere with the right judge on new test samples; $w$ too small, kernel learning will only have memory function but cannot judge new samples[10]. As a result, train samples for the adaptive accelerate of particle swarm optimization (CPSO) to optimize the kernel parameters, in which the parameters set as shown in Table 1. The fitness value and the width of the kernel parameter optimization with the evolutionary process algebra are shown in Fig.2. Through the search you can find a width $w*$ between the values of radial basis function of different faults, as shown in Table 2. We can be seen from Fig.2, the evolution of the fitness function in the 50 generation. The optimal value can be found. For example, a rocky ledge intermediate shaft and gear-tooth fault collapsing, optimal parameter values for 14.1523. Inner bearing on the normal and failure to 0.6176. Normal and intermediate shaft from a rocky ledge, optimal parameter values for the 15.74. These optimal value are the basis for parameter set when carry on KPCA on the gearbox fault feature set.

### 4.3 Feature Extraction Based on KPCA of PSO

Use the raw feature set of 30 groups data at normal state to train in the study, and test with tooth collapse, intermediate shaft float, bearing fault inside and outside the ring, tooth wear and so on 30 groups sample feature set. Compare and analyze the results of linear principal component analysis (PCA) and kernel principal component analysis (KPCA). Fig.3 is the the first and second principal component projection, extracted by PCA and KPCA, of which linear principal component of PCA and non-linear principal component of KPCA are unified dimensionless parameters. It can be seen from figure 3(a) and 3(c) that conventional principal component analysis (PCA) can reduce the dimension of the characteristics of the feature collection, but on the projection of principal component,the principal component feature sample are mixed together and the principal component obtained cannot clearly distinguish the condition type of fault.However,figure 3(b) and 3(d) show that the result obtained by KPCA can clearly distinguish various condition types of fault.Under normal condition the principal component characteristics of data cluster,but the principal component characteristic of inner bearing fault and tooth wear fault are relatively scattered, clearly belonging to different categories.Inner-class spacing of KPCA feature is smaller than PCA,whereas between-class spacing is larger. Principal component extracted from samples after KPCA shows a greater advantage in non-linear analysis,which recognizes the fault condition of gearbox exactly.

## 5 Conclusion

In order to improve the performance of KPCA, a method called KPCA based on PSO is presented and applied in fault condition recognition of gear box. The particle swarm optimization algorithm with adaptive accelerate (CPSO) is applied to optimize the kernel function parameter, which reduces the blindness of KPCA parameter set and can effectively recognize conditions such as normal, tooth wear, tooth collapse and inner ring of bearing. In addition, its fault recognition result is obviously better than PCA. The results show that KPCA based on PSO has advantage in nonlinear feature extraction of mechanical failure, and is helpful for fault conditional recognition of the complicated machine.

## Acknowledgment

The work described in this paper has been supported by National Natural Science Foundation of China under the grant No.50875247 and Province Natural Science Foundation of Shanxi under the grant No.2007011070. The authors would like to express their gratitude for the support of this study.

## References

1. Schoelkopf, B., Smola, A., Mueller, K.R.: Kernel Principal Component Analysis. In: Advances in Kernel Methods-Supper Vector Learning, pp. 327–352. MIT Press, Cambridge (1999)
2. Jongmin, L., Changkyoo, Y., et al.: Nonlinear Process Monitoring Using Principal Component Analysis. Chemical Engineering Science 59, 223–234 (2004)
3. Guanglan, L., Tielin, S., et al.: Gearbox Condition Monitoring Based on Kernel PCA. Mechanical strength 27, 1–5 (2005)
4. Qingbo, H., Fanrong, K., Ruanqiang, Y.: Subspaced-based Gearbox Monitoring by Kernel Principal Component Analysis. Mechanical Systems and Signal Processing 21, 1755–1772 (2007)
5. Ratnaweera, A., Halgamuge, S.K., Watson, H.C.: Self-organizing Hierarchical Particle Swarm Optimizer with Time-varying Acceleration Coefficients. IEEE Transactions on Evolutionary Computation 8, 240–255 (2004)
6. Hongxia, P., Qingfeng, M.: Research on Gearbox Fault Diagnosis Method Based on Adjusting-learning-rate PSO Neural Network. Journal of Donghua University (Eng. Ed.) 23(6), 29–32 (2006)
7. Schoelkopf, B., Smola, A., Mueller, K.R.: Nonlinear principal component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319 (1998)
8. Xuewen, H.: Research on the Theory and Method of Intelligent Fault Diagnosis Based on Support Vector Machine. South university of China, Changsha (2004)
9. Ruixiang, S., Fugee, T., Liangsheng, Q.: Evolving Kernel Principal Component Analysis for Fault Diagnosis. Computer and Industrial Engineer 53, 361–371 (2007)
10. Xinfeng, W., Jing, Q., Guanjun, L.: Kernel Function Optimization in Kernel Principle Component Analysis and Its Application to Feature Extraction of Gear Faults. Journal of Vibration, Measurement & Diagnosis 27(1), 62–64 (2007)

# Evaluation of Distance Measures for Speciated Evolutionary Neural Networks in Pattern Classification Problems

Kyung-Joong Kim[1] and Sung-Bae Cho[2]

[1] Department of Computer Engineering, Sejong University, Seoul, South Korea
[2] Department of Computer Science, Yonsei University, Seoul, South Korea
kimkj@sejong.ac.kr, sbcho@cs.yonsei.ac.kr

**Abstract.** Recently, evolutionary neural networks are hot topics in a neural network community because of their flexibility and good performance. However, they suffer from a premature convergence problem caused by the genetic drift of evolutionary algorithms. The genetic diversity in a population decreases quickly and it loses an exploration capability. Based on the inspiration of diversity in nature, a number of speciation algorithms are proposed to maintain diverse solutions from the population. One problem arising from this approach is lack of information on the distance measures among neural networks to penalize or discard similar solutions. In this paper, a comparison is conducted for six distance measures (genotypic, phenotypic, and behavioral types) with representative speciation algorithms (fitness sharing and deterministic crowding genetic algorithms) on six UCI benchmark datasets. It shows that the choice of distance measures is important in the neural network evolution.

**Keywords:** Evolutionary Neural Networks, Distance Measures, Fitness Sharing, Deterministic Crowding Genetic Algorithm, UCI Benchmark Datasets.

## 1 Introduction

Recently, there is an increasing demand on the distance measures between two neural networks (NN) in evolutionary neural networks (ENN), clustering, and diversity analysis. 1) In evolutionary neural networks, it is important to exploit the relationships among individual neural networks of a population because they are evolved simultaneously. Their similarity or dissimilarity can be used to accelerate evolution for better performance. 2) From a set of neural networks, it is possible to choose representative neural networks by selecting the best one from each cluster. This is helpful to form a selective ensemble instead of one of all neural networks. 3) Diversity prevents learning algorithms from a premature convergence caused by falling into local optimum. The representative applications of distance measures are evolving an ensemble of neural networks [1], robotics, games and pattern classifications.

The distance measures can be categorized into genotypic, phenotypic, and behavioral distances. Each neural network has a topology and weight parameters for each link. The phenotypic distance is defined on the difference of the architectural property. It is

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 630– 637, 2009.

possible to measure distances of two neural networks with their genotypic representations in evolutionary computation. The behavioral distance is measured based on the output of neurons (input, hidden, and output). In this case, there has to be a set of input data to get the outputs. If the neural network is associated to a specific task, it is possible to use their outcome as a source of measuring. In robotics, each NN is used as a controller of robot and the outcomes are actual trajectories. The distance of the two NNs are compared indirectly by comparing two trajectories from different NNs. However, this distance measure is applicable only for the specific domain.

Speciation is a set of techniques to increase the diversity of population in evolutionary algorithms and representatives are fitness sharing and deterministic crowding genetic algorithm [2]. It forces to maintain diverse solutions by punishing over-populated solutions or discarding similar one. The distance measuring is an essential part of the algorithms because they use the closeness of solutions to decide whether you give a penalty or discard one. They are widely used in many applications to overcome the shortcomings of simple evolutionary algorithms and outperform other competitors [3][4]. In evolutionary neural networks, the algorithms are widely used to generate diverse neural networks beneficial to form an ensemble. However, its application in ENN is not limited to the ensemble forming.

It is necessary to compare a number of different distance measures for neural networks because there is no consensus on which one is better than others. The most commonly used one is Euclidean distance of weight vectors because it is the simplest one. There are many alternatives and it is worth to compare them in a unified framework to show their performance difference. In this paper, we categorized known distance measures into three categories (phenotypic, genotypic and behavioral) and compare them in an evolutionary neural network framework. In the evolution, the two representative speciation algorithms are used together with one of distance measures. A good distance measure would lead to increase the diversity of population and avoid local optimums. It results in high performance of the evolutionary neural networks from the last generation.

In behavioral distance measures, it is common to use the difference of the outputs from output neurons, but in this work we devise a new measure that exploits outputs from input/hidden/output neurons. This gives much information on the behavior of the neural networks. There are six distance measures: Euclidean distance of genotypic representation (EU), hamming distance of topology (HA), Euclidean distance of outputs (EU_B), Euclidean of outputs from all neurons (EU_B_ALL), Euclidean distance of average outputs (AO), and Kullback-Leibler entropy of outputs (KL). The proposed methods are tested on six UCI benchmark datasets.

## 2 Background

The simplest way to measure distance is Euclidean distance of genotypic representation. Usually, the chromosome for neural network is a vector of real values and it is easy to use Euclidean distance. It is possible to use more complex representation of neural networks with additional information not just weight. In [6], whole genome is composed of two parts (node genes, genome genes). The connection genome has in-node ID, out-node ID, weight, enable/disable, and original historical ancestor of each gene. In this case, the distance is calculated with a simple linear combination of the matching and mismatching information. In [8][9], authors use the same representation except the historical information. The distance measure is specialized for their representations by exploiting enable/disable information. In [7], a new distance measure is proposed with the consideration of redundancy in representation. Any permutation of the hidden neurons produces the same neural network with a different chromosome representation. $N!$ different representations exist for a network with $n$ hidden neurons. In their work, they consider all permutations in calculating distances of two neural networks. This is computationally expensive because $N!$ combinations are checked.

In behavioral distance measures, the difference of outputs from output neurons is measured with statistical methods. In [5], they propose average output, Pearson correlation and Kullback-Leibler entropy of the outputs. Like genotypic distance measures, Euclidean distance is also used. On the other hands, it is possible to use actual behavior of the neural network to measure the distance. This is dependent on the type of applications of neural networks. If they are used for pattern classification and a robotic problem, the behavior is a vector of class labels classified and trajectories of robot with the neural controller, respectively. Because this is task-dependent, it is not possible to use it for general purpose.

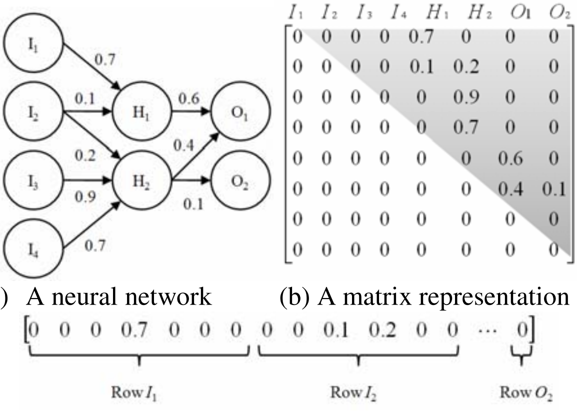
The applications of the distance measures are mainly for evolution because it is a population-based search and there is much interest on the interaction of multiple solutions. [10][11] are exceptional cases and they are not related to evolution. In evolution, the fitness sharing is a dominant application for the distance measures. In many cases, clustering algorithms are used together with speciation and the same distance measures for the speciation are also adopted for the clustering. Interesting applications are diversity analysis of population, distance-based operators, and normalization of solutions.

## 3 Speciated Neural Networks with Distance Measures

A framework for evolving neural networks with speciation is used to compare several distance measures. It evolves topology and weights of neural networks simultaneously. A distance measure is evaluated based on the performance of the best neural network evolved with it. In the speciation, both of the fitness sharing and deterministic crowding algorithms are adopted because the performance of a distance measure could be biased to the choice of a specific speciation method. The best distance measure is decided based on the average performance of the two different speciation methods.

### 3.1 Representation

In this paper, 1-D vector representation is used to represent a neural network. It is a feed-forward neural network with hidden layers. Usually, the neural network can be represented as a matrix whose entry represents the existence of link and its weight. In this representation, only half of the matrix is filled with real-values and another half is not used. The 1-D vector is constructed from the half of the matrix. Figure 1 shows an example of the 1-D vector representation. The shaded area in the matrix representation is converted to the entry of the 1-D vector. If there is a link between two nodes, it is represented with the weight. Otherwise, it is 0.



$$\begin{array}{cccccccc} I_1 & I_2 & I_3 & I_4 & H_1 & H_2 & O_1 & O_2 \\ 0 & 0 & 0 & 0 & 0.7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

(a) A neural network (b) A matrix representation

$$[\underbrace{0\ 0\ 0\ 0.7\ 0\ 0\ 0\ 0}_{\text{Row } I_1}\ \underbrace{0\ 0\ 0.1\ 0.2\ 0\ 0}_{\text{Row } I_2}\ \cdots\ \underbrace{0}_{\text{Row } O_2}]$$

(c) A 1-D vector representation

**Fig. 1.** An example of neural network representation

### 3.2 Speciation Algorithms

Fitness sharing is a method to share its fitness with near neighborhoods. If the population density of the near area of the individual is high, the degree of sharing is increasing and the original fitness of the neural network decreases. The parameter of the algorithm is a sharing radius that decides the boundary of the neighborhood. If the distance between two neural networks is smaller than the radius, they are neighborhood and share fitness. Although there is work on systematic testing on the effect of sharing radius size [14], this is not main issue of our research. In this paper, we set the sharing radius as the half of the average distances of all individuals in an initial population.

Unlike fitness sharing, there is no additional parameter for deterministic crowding genetic algorithm. It is based on the competition between parents and offspring paired to maximize the similarity. Among two similar neural networks, one with better fitness survives to the next generation. Figure 4 summarizes a pseudo code for the deterministic crowding algorithm. It uses the same crossover and mutation with the fitness sharing.

### 3.3 Distance Measures

$N_I$: The number of input neurons
$N_H$: The maximum number of hidden neurons
$N_O$: The number of output neurons
$N$: The number of total neurons ($N$=$N_I$+$N_H$+$N_O$)
$I_{ijk}$, $H_{ijk}$, $O_{ijk}$: $j$th input, hidden, and output neuron of $i$th neural network for $k$th sample
$\varphi$ (): Output of a neuron
$M$: The number of training samples
$V_i$: A chromosome of $i$th neural network (1-D vector)
$L$: A length of the chromosome ($L$= $\frac{(N^2-N)}{2}$)

### 3.3.1 Genotypic Distance

**Euclidean distance of genotypic representation (EU):** This is a simple Euclidean distance of two chromosome vectors.

$$d(a,b)=\sqrt{\sum_{i=1}^{L}(V_{ai}-V_{bi})^2}$$

### 3.3.2 Phenotypic Distance

**Hamming distance of topology (HA):** This counts only the similarity and dissimilarity of topologies. If only one of the networks has a link, this is counted.

$$d(a,b)=\sum_{i=1}^{L}h(V_{ai},V_{bi})$$

$$h(x,y)=\begin{Bmatrix}1 & V_{ai}=0\text{ and }V_{bi}\neq 0, V_{ai}\neq 0\text{ and }V_{bi}=0\\ 0 & \text{otherwise}\end{Bmatrix}$$

### 3.3.3 Behavioral Distance

**Euclidean distance of outputs (EU_B):** This calculates the Euclidean distance of outputs from output neurons.

$$d(a,b)=\sqrt{\sum_{k=1}^{M}\sum_{j=1}^{N_O}(O_{ajk}-O_{bjk})^2}$$

**Euclidean distance of outputs from all neurons (EU_B_ALL):** Usually, the outputs from output neurons are used in the behavioral distance but we propose to use outputs from all neurons (input/hidden/output neurons).

$$d(a,b)=\sqrt{\sum_{k=1}^{M}(\sum_{j=1}^{N_I}(I_{ajk}-I_{bjk})^2+\sum_{j=1}^{N_H}(H_{ajk}-H_{bjk})^2+\sum_{j=1}^{N_O}(O_{ajk}-O_{bjk})^2)}$$

**Euclidean distance of average outputs (AO):** This is the Euclidean distance of average outputs from each output neuron.

$$d(a,b)=\sqrt{\sum_{j=1}^{N_O}(\overline{O}_{aj}-\overline{O}_{bj})^2}$$

$$\overline{O}_{aj}=\frac{1}{M}\sum_{k=1}^{M}O_{ajk}\quad \overline{O}_{bj}=\frac{1}{M}\sum_{k=1}^{M}O_{bjk}$$

**Kullback-Leibler entropy of outputs (KL):** This is proposed in [5]. It is called as relative entropy and modified to be used as a distance measure.

$$d(a,b)=\frac{1}{2}\sum_{k=1}^{M}\sum_{j=1}^{N_O}(O_{ajk}\log\frac{O_{ajk}}{O_{bjk}}+O_{bjk}\log\frac{O_{bjk}}{O_{ajk}})$$

## 4 Experimental Results

The proposed method is evaluated on six UCI benchmark datasets. Table 1 summarizes statistics of the datasets. The number of classes is 2 or 3. The number of attributes is ranged from 4 to 34. The number of samples is from 351 to 1473. In case of a missing entry, it is imputed randomly based on the probability of other entry values. The attribute value is normalized to 0~1. The whole data is separated to training (2/3) and test (1/3) randomly. Table 2 summarizes parameters used in this experiment. The number of input and output neurons of a neural network is the same with the number of attributes and classes of a dataset, respectively. A logistic function is used as a transfer function of the neural network. The result is averaged over 10 runs (each time, the training/test samples are randomly separated). For each run, a seed for random functions is decided based on the current time and the value is used for 12 combinations of settings (2 speciation algorithms × 6 distance measures). The 12 variations use the same initial population. The fitness function is classification accuracy on the training samples. They are compared in terms of the accuracy on the test samples.

**Table 1.** Summary of dataset statistics

| Name | # of classes | # of attributes | # of samples | # of training samples | # of test samples |
|---|---|---|---|---|---|
| Balance | 3 | 4 | 625 | 417 | 208 |
| Breast Cancer | 2 | 9 | 699 | 466 | 233 |
| Contraceptive Method Choice | 3 | 9 | 1473 | 982 | 491 |
| Congressional Voting Record | 2 | 16 | 435 | 290 | 145 |
| Ionosphere | 2 | 34 | 351 | 234 | 117 |
| Tic-Tac-Toe | 2 | 9 | 958 | 639 | 319 |

**Table 2.** Parameters of experiments

| Name | Value |
|---|---|
| Population Size | 50 |
| The Maximum Number of Generation | 500 |
| Crossover Rate | 0.9 |
| Mutation Rate | 0.1 |
| The Number of Maximum Hidden Nodes | 15 |

Table 3 summarizes the experimental results of the 6 different distance measures with the fitness sharing algorithm and deterministic crowding algorithm. It shows that the best one (in terms of average accuracy) is different for each dataset and speciation method. Based on the average of 12 cases (2 speciation methods × 6 datasets) EU_B_ALL and AO are the best, and HA is the worst.

**Table 3.** Summarization of results

| | EU | HA | EU_B | EU_B_ALL | AO | KL |
|---|---|---|---|---|---|---|
| Balance | **87.5 ± 3.0** | 85.5 ± 4.3 | 84.1 ± 3.3 | 87.1 ± 2.3 | 83.3 ± 4.6 | 82.4 ± 4.7 |
| Breast Cancer | 95.8 ± 1.1 | **95.9 ± 1.4** | 95.4 ± 1.8 | 95.7 ± 0.9 | 94.7 ± 1.1 | 94.9 ± 1.4 |
| Contraceptive | 43.8 ± 2.6 | 45.7 ± 1.8 | 44.5 ± 2.1 | 44.4 ± 1.8 | 45.2 ± 1.9 | **46.1 ± 2.6** |
| Congressional Voting | 91.0 ± 3.0 | 91.2 ± 2.2 | 91.8 ± 3.1 | **93.0 ± 2.2** | 92.8 ± 3.6 | 90.8 ± 3.9 |
| Ionosphere | 80.8 ± 4.7 | **84.2 ± 3.6** | 81.0 ± 4.2 | 81.7 ± 4.8 | 83.3 ± 5.4 | 81.7 ± 2.6 |
| Tic-Tac-Toe | 68.2 ± 3.6 | 68.9 ± 2.6 | **70.6 ± 3.0** | 69.6 ± 3.0 | 70.5 ± 1.5 | 70.2 ± 3.2 |

(a) Fitness sharing

| | EU | HA | EU_B | EU_B_ALL | AO | KL |
|---|---|---|---|---|---|---|
| Balance | 85.3 ± 2.9 | 78.8 ± 4.7 | **87.6 ± 1.2** | 86.2 ± 2.3 | 86.6 ± 2.6 | 87.4 ± 2.4 |
| Breast Cancer | 95.5 ± 0.8 | 95.4 ± 1.6 | 95.7 ± 1.3 | **95.9 ± 1.1** | 95.6 ± 1.4 | 95.4 ± 1.0 |
| Contraceptive | 46.2 ± 1.1 | 43.9 ± 1.8 | 45.8 ± 1.4 | 45.8 ± 2.0 | 46.2 ± 1.6 | **46.6 ± 1.7** |
| Congressional Voting | 92.8 ± 2.2 | 90.4 ± 4.0 | 93.6 ± 2.5 | 93.5 ± 2.0 | **93.8 ± 1.6** | 93.7 ± 2.5 |
| Ionosphere | 85.3 ± 3.7 | 83.6 ± 3.4 | **86.2 ± 2.5** | 85.6 ± 1.9 | 85.0 ± 4.1 | **86.2 ± 2.5** |
| Tic-Tac-Toe | **71.0 ± 2.4** | 69.1 ± 2.2 | 70.9 ± 1.0 | 69.7 ± 2.3 | 70.9 ± 1.5 | 70.8 ± 1.8 |

(b) Deterministic crowding genetic algorithm

| | EU | HA | EU_B | EU_B_ALL | AO | KL |
|---|---|---|---|---|---|---|
| Average | 78.6 | 77.7 | 78.9 | **79.0** | 79.0 | 78.9 |

(c) Average results

## 5 Conclusion

In this paper, the distance measures for neural networks are compared in the framework of evolutionary neural networks with speciation. Because the evolutionary system is dependent on how good the distance measures are, several candidates can be evaluated indirectly based on the performance of the final neural network evolved. For more accurate evaluation, two representative speciation algorithms are simultaneously used with six UCI benchmark datasets.

Based on the categorization of distance measures, we compare six distance measures from each group (genotypic, phenotypic and behavioral). In the behavioral distance measure, new one called EU_B_ALL is proposed considering outputs from all neurons (input/hidden/output). The experimental results show that behavioral distance measures outperformed than genotypic and phenotypic distance measures.

## Acknowledgements

Co-author Kyung-Joong Kim was supported by Korea Health 21 R&D Project, Ministry for Health, Welfare and Family Affairs (A040163).

## References

[1] Yao, X., Islam, M.M.: Evolving artificial neural network ensembles. IEEE Computational Intelligence Magazine (2008)

[2] Baeck, T., Fogel, D.B., Michalewicz, Z.: Advanced algorithms and operations (Evolutionary Computation). Taylor & Francis, London (2000)

[3] Hwang, K.-S., Cho, S.-B.: Improving evolvable hardware by applying the speciation technique. Applied Soft Computing 9(1), 254–263 (2008)

[4] Hong, J.-H., Cho, S.-B.: Efficient huge-scale feature selection with speciated genetic algorithm. Pattern Recognition Letters 27(2), 143–150 (2006)

[5] Ahn, J.-H., Cho, S.-B.: Speciated neural networks evolved with fitness sharing technique. In: Proceedings of the 2001 Congress on Evolutionary Computation, pp. 390–396 (2001)

[6] Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation 10(2), 99–127 (2002)

[7] Jung, S.-C., Moon, B.-R.: Central point crossover for neuro-genetic hybrids. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 1292–1303. Springer, Heidelberg (2004)

[8] Sallam, H., Regazzoni, C.S., Talkhan, I., Atiya, A.: Measuring the genotype diversity of evolvable neural networks. In: Proceedings of the 6th International Conference on Informatics and Systems, pp. 38–43 (2008)

[9] Sallam, H., Regazzoni, C.S., Talkhan, I., Atiya, A.: Evolving neural networks ensembles NNEs. In: 2008 IAPR Workshop on Cognitive Information Processing, pp. 142–147 (2008)

[10] Ormoneit, D.: A regularization approach to continuous learning with an application to financial derivatives pricing. Neural Networks (12), 1405–1412 (1999)

[11] Li, K., Huang, H.-K., Ye, X.-C., Cui, L.-J.: A selective approach to neural network ensemble based on clustering technology. In: Proceedings of the Third International Conference on Machine Learning and Cybernetics, pp. 3229–3233 (2004)

[12] Khare, V., Yao, X.: Artificial speciation of neural network ensembles. In: Proceedings of the 2002 UK Workshop on Computational Intelligence, pp. 96–103 (2002)

[13] Trujillo, L., Olague, G., Lutton, E., de Vega, F.F.: Discovering several robot behaviors through speciation. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., McCormack, J., O'Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, A.Ş., Yang, S. (eds.) EvoWorkshops 2008. LNCS, vol. 4974, pp. 164–174. Springer, Heidelberg (2008)

[14] Cioppa, A.D., Stefano, C.D., Marcelli, A.: On the role of population size and niche radius in fitness sharing. IEEE Transactions on Evolutionary Computation 8(6), 580–592 (2004)

# Emergence of Different Mating Strategies in Artificial Embodied Evolution

Stefan Elfwing, Eiji Uchibe, and Kenji Doya

Neural Computation Unit, Okinawa Institute of Science and Technology,
12-22 Suzaki, Uruma, Okinawa 904-2234, Japan
{elfwing,uchibe,doya}@oist.jp

**Abstract.** In nature, discrete alternative mating strategies have been observed in many different species. In this paper, we investigate the emergence of different mating strategies in a small colony of simulated robots, using our previously proposed framework for performing embodied evolution with a limited number of robots. The virtual agents can reproduce offspring by mating, i.e., an exchange of genotypes with another robot. In the experiments, we observed two individual mating strategies: 1) *Roamer strategy*, where the agents never wait for potential mating partners; and 2) *Stayer strategy*, where the agents wait for potential mating partners depending on their internal energy level, the distance to the mating partner's tail-LED, and the distance to the closest battery. The most interesting finding was that in some simulations the evolution produced a mixture of mating strategies within the population, typically with a narrow roamer subpopulation and a broader stayer subpopulation with distinct differences in genotype, phenotype, behavior, and performance between the subpopulations.

## 1 Introduction

In nature, discrete alternative mating strategies have been observed in many different species. The alternative reproductive tactics are either correlated with genotype, phenotype, density, environment, or social context (see e.g., [1,2]). In this study, we investigate the emergence of different mating strategies in a small colony of simulated Cyber Rodent robots [3], using our previously proposed framework for performing embodied evolution [4] with a limited number of robots [5,6]. The framework utilizes time-sharing in subpopulations of virtual agents hosted in each robot. The virtual agents can reproduce offspring by mating, i.e., an exchange of genotypes with another robot.

In the experiment, we observed two individual mating strategies: 1) *Roamer strategy*, where the agents never wait for potential mating partners; and 2) *Stayer strategy*, where the agents wait for potential mating partners depending on their internal energy level, the distance to the mating partner's tail-LED, and the distance to the closest battery. Three main types of population mating strategies, i.e., distributions of individual mating strategies in the populations, were obtained in different simulation runs: 1) *Roamers*, where all, or almost all, of

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 638–647, 2009.

the individuals are roamers; 2) *Stayers*, where there is an unimodal distribution of waiting thresholds; and 3) *Mixture of strategies*, typically with a narrow roamer subpopulation and a broader stayer subpopulation with distinct differences in genotype, phenotype, behavior, and performance between the subpopulations.

The research is an attempt to use artificial evolution to investigate the mechanisms and conditions for the emergence of a real biological phenomena, i.e., the emergence of different mating strategies in many species. Earlier research in the same vein was conducted by Floreano *et al.* [7]. They used a small colony of mobile robots to investigate the evolutionary conditions conducive to the emergence of communication. In their work the communication abilities of the robots were directly dependent on the evolutionary selection regimes, which were predefined by the authors. In contrast, in our work the emergence of different mating strategies between and within simulations occur under homogenous evolutionary conditions.

## 2 Method

Fig. 1 shows an overview of our earlier proposed embodied evolution framework [5,6]. In the framework, the population consists of $N_{sub}$ robots, each of which contains a subpopulation of $N_{va}$ virtual agents. The virtual agents are evaluated, in random order, by time-sharing, i.e., taking control over the robot for a limited number of $T_{ts}$ time steps. There is a random selection of potential offspring, which are produced in proportion to the reproductive ability of the parental virtual agents. Each subpopulation contains a list, $\boldsymbol{O}$, of potential offspring reproduced during the current generation. After all virtual agents in a subpopulation have survived for a full lifetime or died a premature death, a new subpopulation is created by randomly selecting $N_{va}$ potential offspring from $\boldsymbol{O}$.

A virtual agent that controls a robot can reproduce potential offspring by performing a reproductive mating, which consists of a pair-wise exchange of genotypes with a virtual agent controlling another robot (hereafter called a mating) and also by satisfying a predefined reproduction condition (hereafter called a reproductive mating). For example, in our experiments the probability of reproducing potential offspring is proportional to the virtual agent's internal energy at the mating occasion. This means that the reproductive success of a virtual agent is a combination of the virtual agent's ability to perform genotype exchanges and the ability to keep its internal energy level high, by recharging from external batteries in the environment. A virtual agent that performs a reproductive mating creates two potential offspring that are appended to the list of potential offspring $\boldsymbol{O}$ in the agent's subpopulation, by applying genetic operations to the agent's genotype and the genotype received from the mating partner.

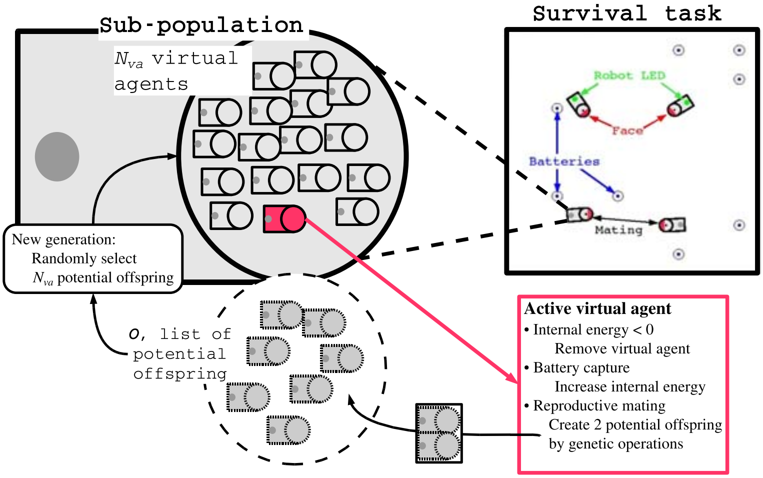


**Fig. 1.** Overview of the embodied evolution framework. Each robot contains a subpopulation of virtual agents of size $N_{va}$, who are evaluated by time-sharing, i.e., taking control over the robot for a limited period of time. If a virtual agent performs a reproductive mating, it creates two potential offspring by applying genetic operations to the virtual agent's own genotype and the genotype received from the mating partner. The two potential offspring are then inserted in the list of potential offspring, $\boldsymbol{O}$, of the subpopulation containing the virtual agent. When all virtual agents in a subpopulation have either survived for a full lifetime or died a premature death, a new subpopulation is created by randomly selecting $N_{va}$ potential offspring from $\boldsymbol{O}$.

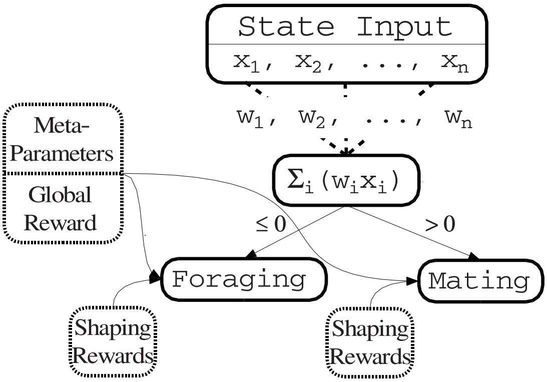


**Fig. 2.** Overview of the two-layered control architecture used in the experiments. The top-layer linear feed-forward neural network uses the $n$-dimensional state space as input. In each time step, the neural controller selects one of the two reinforcement learning modules, foraging or mating, The selected module then executes an action based on its learned behavior.

The virtual agents are controlled by a two-layered control architecture (see Fig. 2). In each time step, an evolutionary tuned linear feed-forward neural network selects one of two reinforcement learning modules, foraging or mating. The selected module then executes an action based on its learned behavior. It also updates its learning parameters, based on a global reward function, the meta-parameters (shared by all modules and genetically determined), and its shaping reward function (unique for each module and genetically determined). In the experiments, the global reward for the reinforcement learning modules is set to +1 for a mating and a battery capture, otherwise the reward is set to 0. The basic behaviors are learned by the Sarsa reinforcement learning algorithm [8,9] with tile coding [9] and potential-based shaping rewards [10]. For a detailed description of our framework and implementation details see [5,6].

## 3 Experiments

In our earlier work [5,6], we analyzed the overall results of the proposed embodied evolution framework. In this paper, we focus our analysis on the different mating strategies that emerged from the embodied evolution process. A mating strategy is determined by the neural network which in each time step selects if a virtual agent tries to reproduce offspring by mating with another robot or if it tries to recharge its internal battery by capturing external batteries.

In the experiments, we used a simulation environment (see the figure used to represent the survival task in Fig. 1) that was developed to mimic the properties of the Cyber Rodent robot [3]. In the experiment, we used four robots that can detect each other by the green tail-LEDs, and by the red faces in the front of the robots, where the infrared ports for genotype exchanges are located. Inside each robot there was 20 virtual agents and each virtual agent was evaluated for three time-sharings of 400 time steps, giving a total lifetime of 1200 time steps. We performed experiments with 4, 6, 8, 10, 12, 14, and 16 batteries in the environment and for each battery condition, we performed 10 simulations running for 1000 generations. In the experiments, a virtual agent has a maximum internal energy level ($E_{max}$) of 500 energy units. Each time step, the energy level decreases by 1 energy unit and a battery capture increases the energy level by 100 energy units. If a virtual agent's energy ($E$) is depleted ($E \leq 0$), then the virtual agent dies and it is removed from its subpopulation. The probability ($p_{mate}$) that a virtual agent reproduces potential offspring, i.e., a mating becomes a reproductive mating, is linear dependent on the virtual agent's internal energy level: $p_{mate} = E/E_{max}$. This creates an interesting trade-off, where the reproductive success of a virtual agent depends both on the agent's ability to perform matings and the ability to maintain a high energy level by capturing batteries to increase the probability to reproduce potential offspring at the mating occasions. In the experiments, there is no explicit fitness function. Instead, we consider the fitness to be equal to the number of reproductive matings, i.e., the number of times a virtual agent reproduces potential offspring.

### 3.1 Obtained Mating Strategies

The output of the linear neural network is equal to $\sum_i w_i x_i$, where $x_i$ is the $i$th input with the corresponding genetically determined weight $w_i$. If the output is greater than zero then agent selects the mating behavior, otherwise the agent selects the foraging behavior. The input to the neural network consists of a bias of 1 ($x_1$), the virtual agent's normalized internal energy ($x_2$), and the normalized distances to the closest battery ($x_3$), the closet tail-LED ($x_4$), and the closest face of another robot ($x_5$). If a target was not visible, then the corresponding input value was set to $-1$.

The input space can be divided into five state types: 1) $S_1$, when the face of another robot is visible; 2) $S_2$, when only a battery is visible; 3) $S_3$, when a battery and a tail-LED are visible; 4) $S_4$, when only a tail-LED is visible; and 5) when no visible information is available. In four of the five state types $S_1$, $S_2$, $S_4$, and $S_5$ almost all virtual agents in the last 20, out of 1000, generations of all simulation runs in all battery conditions obtained the same simple policy. If a face is visible, then the mating behavior is always selected. Otherwise, the foraging behavior is always selected. The interesting case is the third state type, $S_3$, i.e., whether the virtual agent tries to capture the battery and look for a better mating opportunity later, or if the virtual agent waits for the potential mating partner to turn around and show its face. The neural network selection policy in $S_3$ can be expressed as energy thresholds of mating, $E_m$ for different normalized distances to the closest battery ($x_3$) and the closest tail-LED ($x_4$):

$$E_m = -(w_1(1) + w_3 x_3 + w_4 x_4 + w_5(-1))/w_2. \quad (1)$$

Here, a virtual agent selects the mating behavior if the agent's normalized internal energy is greater than $E_m$, otherwise the agent selects the foraging behavior. In the computations of the threshold values, the values were limited to the possible range of energy levels (normalized to $[0, 1]$), i.e., if the "raw" threshold

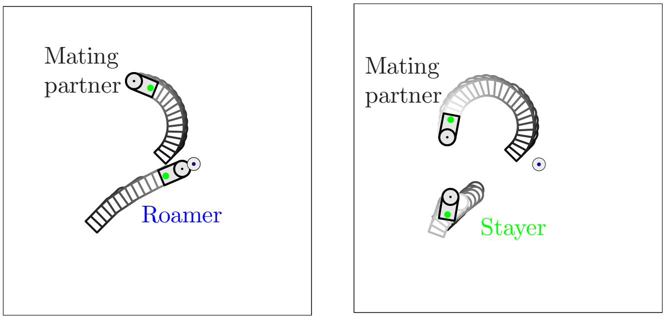


**Fig. 3.** Example trajectories of the mating strategies of a roamer (left) and a stayer (right) in $S_3$. The roamer ignores the tail-LED of the mating partner and selects the foraging behavior to capture the battery pack. The stayer selects the mating behavior and adjusts its position according to the trajectory of the mating partner.

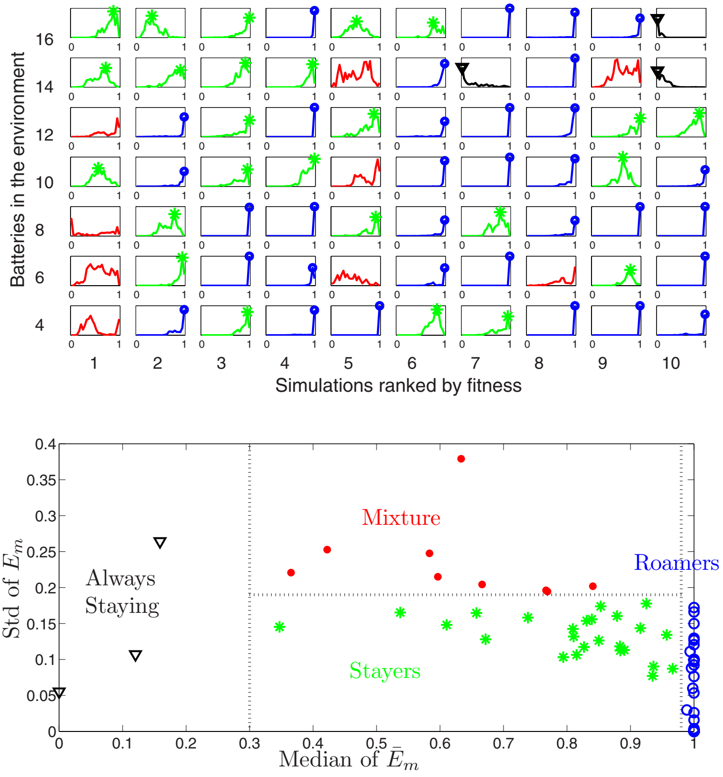


**Fig. 4.** The obtained distributions of $\bar{\mathbf{E}}_m$ in all simulations (top figure) and the classification of different population mating strategies using the median and the standard deviation of $\bar{\mathbf{E}}_m$ (bottom figure)

value was larger (smaller) than the maximum (minimum) energy level then the threshold value was assigned to the maximum (minimum) energy level. To be able to compare the obtained policies in $S_3$, we estimated the average energy threshold, $\bar{\mathbf{E}}_m$, by computing the average value over 36 sample inputs.

In the simulations, two types of individual mating strategies[1] were obtained (illustrated in Fig. 3): 1) *Roamer strategy* with the $\bar{\mathbf{E}}_m$-value equal to the maximum threshold (1), which means that, for all state types, roamers always select mating if a face is visible, otherwise they always select foraging; and 2) *Stayer strategy* with the $\bar{\mathbf{E}}_m$-value lower than the maximum threshold and where the decision of whether to select mating or foraging depends on the distance to the battery, the distance to the tail-LED, and the current energy level.

To illustrate the obtained population mating strategy in the simulations, we computed the distributions of average mating thresholds in $S_3$, $\bar{\mathbf{E}}_m$, for all virtual

[1] The terminology roamers and stayers is borrowed from Sandell *et al.* [11].

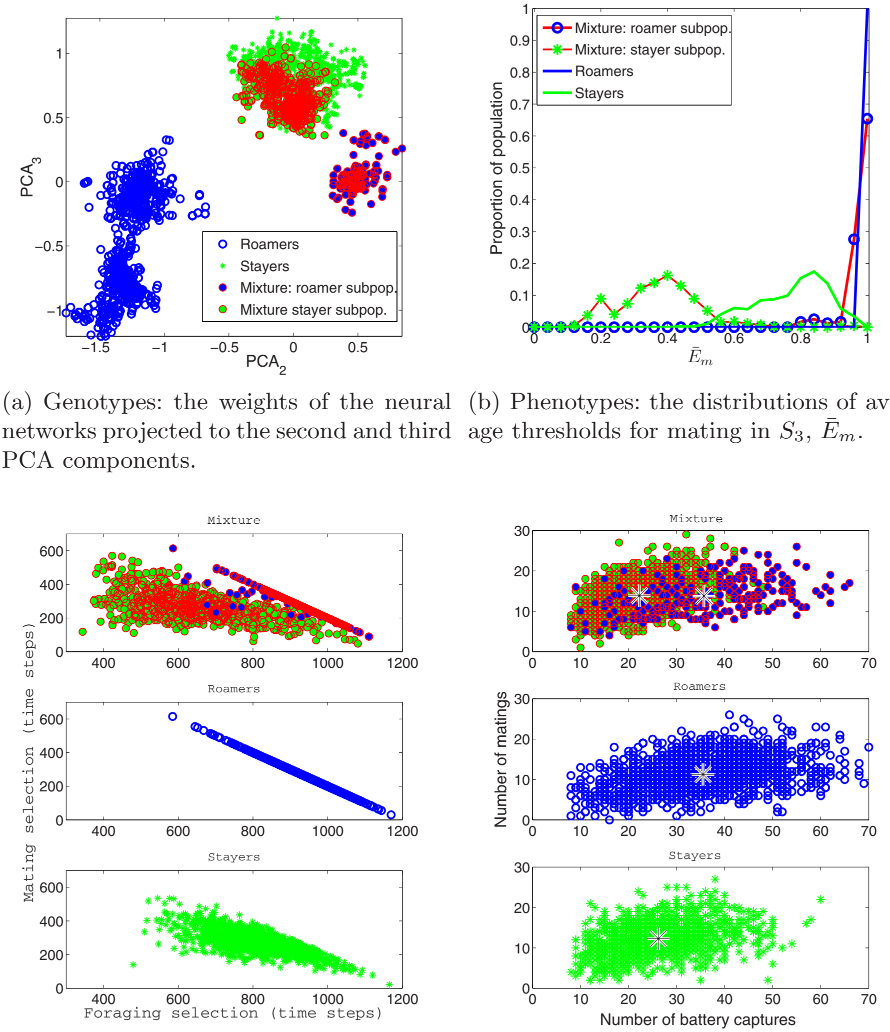
agents in final 20 generations. Fig. 4(a) shows the distributions of the $\bar{\mathbf{E}}_m$-values in all battery conditions in all simulations. In each condition, the simulations are ranked by average fitness and the fittest simulations are shown in the leftmost column. The color coding and markers represent the four types of population mating strategies obtained in the simulation and they are classified according to the median and standard deviation of $\bar{\mathbf{E}}_m$ (Fig. 4(b)): 1) *Roamers* (blue curves with circles at the distribution peaks, where the median of $\bar{\mathbf{E}}_m > 0.98$): for all, or for a majority of, the virtual agents the average mating threshold is equal to the maximum energy level; 2) *Always staying* (black curves with triangles and the distribution peaks, where the median of $\bar{\mathbf{E}}_m < 0.3$): there is a narrow distribution with a majority of the virtual agents having an average threshold equal to, or close to, the minimum threshold; 3) *Stayers* (green curves with asterisks at the distribution peaks, where the standard deviation of $\bar{\mathbf{E}}_m < 0.19$): there is a unimodal distribution of the average energy thresholds with the peak of the distribution lower than the maximum threshold; and 4) *Mixture of strategies* (red curves where the standard deviation of $\bar{\mathbf{E}}_m > 0.19$): there is a multimodal distribution of average thresholds, typically with a narrow roamer subpopulation and a broader stayer subpopulation.

Interestingly, the more cooperative and heterogenous population strategies (i.e., stayers and the mixture of strategies) perform better than the more selfish and homogenous roamers. For example, in all environments the best performing simulations either obtained the stayers strategy or the mixture of strategies. Also, in the best performing half of the simulations in all environments 69% (24 out of 35) obtained the one of two more cooperate strategies. In contrast, in the worst performing half of the simulations only 31% (10 out of 32) obtained a more cooperate strategy (excluding the three low-performing simulations obtaining the always staying strategy).

### 3.2 Analysis of the Main Population Mating Strategies

To analyze the three main obtained population mating strategies (roamers, stayers, and mixture of strategies), we investigated the relationship between genotype, phenotype, behavior, and performance for populations strategies obtained in the four battery condition (simulations ranked 1, 5, and 6 in the bottom row of Fig. 4(a)). These relationships are illustrated in Fig. 5, where the data for all virtual agents that survived (their internal energy levels were positive for the full 1200 time steps lifetime) in the final 20 generations are shown. For the mixture of strategies, the two clusters in the PCA-space (Fig. 5(a)) were used to identify two separate genetic traits in the population. The two subpopulation representing the two traits are color-coded according to their phenotype-distributions (Fig. 5(b)): red and blue colors for the roamer subpopulation and red and green colors for the stayer subpopulation.

The figures shows clear differences between the population mating strategies, where differences in the average energy threshold distributions (directly determined by the genotype distributions) lead to significant differences in the number of time steps the virtual agents select the foraging behavior (see Fig. 5(c)), the



(a) Genotypes: the weights of the neural networks projected to the second and third PCA components.

(b) Phenotypes: the distributions of average thresholds for mating in $S_3$, $\bar{E}_m$.

(c) Behaviors: time steps mating was selected (mating behavior when a face was visible) plotted against time steps the foraging behavior was selected.

(d) Performance: the number of matings plotted against the number battery captures.

**Fig. 5.** Illustrations of the relationship between genotypes (top left), the phenotypes (top right), behaviors (bottom left), and performance (bottom right) for the three main obtained population mating strategies

time steps the virtual agents select waiting for a potential mating partner (mating behavior when no face is visible), and the number of captured batteries (see Fig. 5(d), where the big asterisks indicate the population averages). For examples, the roamer subpopulation in the mixture of strategies spends, on average, 77% of its lifetime foraging and only 0.5% waiting. In contrast, the stayer subpopulation spends 53% of its lifetime foraging and 24% waiting. The roamer subpopulation captures, on average 35.7 batteries, compared with 22.3 batteries for the stayer subpopulation. The results related to mating are remarkably similar for all strategies, both for the distribution of the number time steps of mating selection (Fig. 5(c)) and the distribution of the number of performed matings (Fig. 5(d)).

The mixture of strategies is the most interesting of the obtained population strategies, because the differences in behaviors and phenotypes within the population can be traces back to two distinct genetic traits which are "alive" in population at the same time. To investigate the stability of the mixture of the roamer- and stayer subpopulations, we performed an additional experiment where we measured the average fitness for different proportions of the number of virtual agents of the two subpopulations. These results are shown in Fig. 6, as the average fitness of the roamers and stayers as a function of the proportion of roamers in the population. In the experiment, the roamers and the stayers were randomly selected from the final 20 generations of the evolutionary experiments (identified by the neural network weights part of their genotypes, as shown in Fig. 5(a)). Interestingly, the mixture of the roamers and stayers subpopulations constitute an evolutionary stable state, as seen by that the average fitness curves intersect at a proportion of approximately 30% roamers in the population. For the roamers, the average fitness increases if there are fewer roamers in the population, while the opposite is true for stayers. This gives a dynamic where more

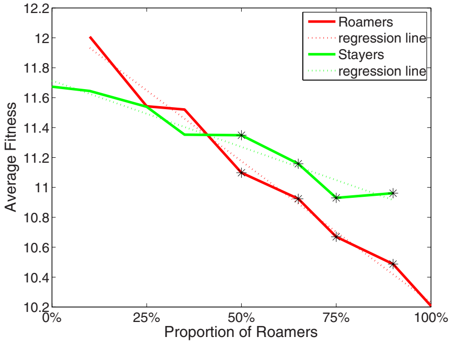


**Fig. 6.** Average fitness of the roamer subpopulation and the stayer subpopulation as a function of the proportion of roamers in the population. The black asterisks indicate subpopulation ratios for which there was a significant, $p = 0.05$, difference in average fitness (over 100 simulation runs) between the two subpopulations.

roamers (compared to the stable proportion) lead to lower average fitness of the whole population and that the stayers will, on average, produce more offspring than the roamers, which will pull back the proportion of roamers toward the stable state. More stayers lead to higher average fitness of the whole population and that the roamers will produce more offspring, which will again pull the the proportion of roamers toward the stable state.

## 4 Conclusions

In this paper we show that different mating strategies can emerge in a small colony of simulated robots. The most interesting finding was that in some simulations the evolution produced a mixture of mating strategies, typically with a roamer subpopulation and a stayer subpopulation with distinct differences in genotype, phenotype, behavior, and performance between the subpopulations.

## References

1. Bleay, C., Comendant, T., Sinervo, B.: An experimental test of frequency-dependent selection on male mating strategy in the field. Proc. Biol. Sci. 274, 2019–2025 (2007)
2. Brockmann, J.H.: An experimental approach to altering mating tactics in male horseshoe crabs (limulus polyphemus). Behavioral Ecology 13(2), 232–238 (2002)
3. Doya, K., Uchibe, E.: The cyber rodent project: Exploration of adaptive mechanisms for self-preservation and self-reproduction. Adaptive Behavior 13(2), 149–160 (2005)
4. Watson, R., Ficici, S., Pollack, J.: Embodied evolution: Distributing an evolutionary algorithm in a population of robots. Robotics and Autonomous Systems 39(1), 1–18 (2002)
5. Elfwing, S., Uchibe, U., Doya, K., Christensen, H.I.: Darwinian embodied evolution of the learning ability for survival. Adaptive Behavior (2009) (in Press)
6. Elfwing, S., Uchibe, E., Doya, K.: Co-evolution of rewards and meta-parameters in embodied evolution. In: Sendhoff, B., Körner, E., Sporns, O., Ritter, H., Doya, K. (eds.) Creating Brain-Like Intelligence. LNCS, vol. 5436, pp. 278–302. Springer, Heidelberg (2009)
7. Floreano, D., Mitri, S., Magnenat, S., Keller, L.: Evolutionary conditions for the emergence of communication in robots. Current Biology 17(6), 514–519 (2007)
8. Rummery, G.A., Niranjan, M.: On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University Engineering Department (1994)
9. Sutton, R.S.: Generalization in reinforcement learning: Successful examples using sparse coarse coding. In: Proceedings of Advances in Neural Information Processing Systems (NIPS 1996), vol. 8, pp. 1038–1044 (1996)
10. Ng, A., Harada, D., Russell, S.: Policy invariance under reward transformations: Theory and application to reward shaping. In: Proceedings of the International Conference on Machine learning (ICML 1999), pp. 278–287 (1999)
11. Sandell, M., Liberg, O.: Roamers and stayers: A model on male mating tactics and mating systems. The American Naturalist 139(1), 177–189 (1992)

# A Markov Model for Multiagent Patrolling in Continuous Time

Jean-Samuel Marier, Camille Besse, and Brahim Chaib-draa

Department of Computer Science and Software Engineering, Laval University, Quebec, Canada
{marier,besse,chaib}@damas.ift.ulaval.ca

**Abstract.** We present a model for the multiagent patrolling problem with continuous-time. An anytime and online algorithm is then described and extended to asynchronous multiagent decision processes. An online algorithm is also proposed for coordinating the agents. We finally compared our approach empirically to existing methods.

**Keywords:** Multiagent, Unmanned Autonomous Vehicle, Online, Patrol.

## 1 Introduction

Unmanned Aerial Vehicles (UAVs) are a promising technology for many information gathering applications. As these devices become cheaper, more robust and have increased autonomy, we can expect to see them used in many different new applications such as surveillance and patrol missions. In such missions, the status of some sites must be monitored for events. If an UAV must be close from a location to monitor it correctly and the number of UAV does not allow covering each site simultaneously, a path planning problem arises: how should the agents visit the locations in order to make sure that the information about all locations is as accurate as possible?

In the last decade, several patrolling algorithms have been developed. For instance, Santana *et al.* [1] proposed a graph patrolling formulation on which agents use reinforcement learning on a particular Markov Decision Process (MDP). They defined this MDP over a countably infinite state space, leading to long and costly computations. Their approach also assumes that agents communicate by leaving messages on the nodes of the graph, leading to unrealistic communication models. On the other hand, reactive algorithms such as the ant-colony approach from [2] have been shown to perform well in theory as well as empirically. However such approach also rely on simplistic communication models relying on so-called pheromones.

In the case where all locations are equally important, Chevaleyre [3] proved that the shortest Hamiltonian circuit is an optimal solution for a single agent and a complete empirical study [4] shows that multiagent strategies using a unique cycle are the best whatever the graph is. However, as some locations may be more important than others, not visiting the less important ones from time to time may be advantageous.

In this paper, we propose a new graphical formulation of the patrolling problem featuring uncertainty on sensors and on information retrieval located on nodes of the graph and importance of certain locations over others. Contrary to previous approaches

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 648–656, 2009.

that tried to minimize the idleness of the vertices of the graph, our model outlines the information retrieval aspect of the patrolling problem and the decay of the information value as the age of the information retrieved increases, forcing the agent to update its knowledge about nodes as frequently as possible. Our approach has a continuous-time formulation, allowing real durations and asynchronicity and it is readily usable for scenarios where durations are uncertain. We also present a planning algorithm that is suitable for the proposed formulation.

The remainder of this paper is structured as follows: section 2 presents the proposed model for patrol problems, section 3 presents the multiagent MDP framework and a mapping of the problem at hand as a continuous-time MDP, an online algorithm for solving the problem is presented in section 4, and section 5 presents experiments. Finally sections 6 and 7 discusses results and future work.

## 2 Problem Formulation

The patrolling problem has a graphical structure. Let us use $V$ for the vertex set of that graph and $E$ for its edge set. Let $L$ be an $|V| \times |V|$ matrix, in which $L_{ij}$ is a real number that represents the time required to go travel from $i$ to $j$ if $[i, j] \in E$ and is infinite otherwise. Each vertex $i$ has a real *non-negative* importance weight, noted $w_i$. We note $\mathbf{w}$ the vector of all such weights.

In patrolling literature such as [2], idleness is used as a performance measure. The idleness of vertex $i$, noted $\tau_i$ represents the time since the last visit of an agent to that vertex. The idleness is 0 if and only if an agent is currently at vertex $i$ and $\tau_i^{t+\Delta t} = \tau_i^t + \Delta t$ if there are no visits to $i$ in the time interval $(t, t + \Delta t)$. Because idleness is an unbounded quantity, a more suitable representation is to use $k_i^t = b^{\tau_i^t}$, with $0 < b < 1$. We call this exponential idleness. Since $k_i^t$ is always in $[0, 1]$, it can be seen as the expected value of a Bernoulli random variable which has value 1 if vertex $i$ is observed correctly and 0 otherwise. Thus, $k_i^t$ is the probability that this random variable is 1 at time $t$.

The probability evolves as $k_i^{t+\Delta t} = k_i^t b^{\Delta t}$ if there are no visits to $i$ during time interval $(t, t+\Delta t)$. If an agent with noisy observations visits $i$ at time $t$, idleness becomes 0 with probability $b < (1 - a) \leq 1$. If $n$ agents visit vertex $i$ at time $t + \Delta t$ and that there were no visits since time $t$:

$$k_i^{t+\Delta t} = k_i^t a^n b^{\Delta t} + 1 - a^n. \tag{1}$$

To sum up, an instance of the patrolling problem is a tuple $\langle L, \mathbf{w}, a, b\rangle$, consisting respectively of the matrix $L$ of edge lengths, the vector $\mathbf{w}$ of importance weights and parameters $a$ (the probability that the idleness does not become 0 when an agent visits a vertex) and $b$ (the rate at which $k_i$ decays over time).

## 3 Multiagent Markov Decision Processes (MMDPs)

This section casts the previous problem as a Multiagent MDP (MMDP). We assume that the problem state is fully observable, i.e. every agent has the same complete information to make its decision. Such problems are called MMDPs. In the patrolling problem however, the actions of each agent have a concurrent effect on the the environment and they are of different durations. Concurrency in decision processes is conveniently modeled with a Generalized Semi-Markov Decision Process (GSMDP), introduced by Younes *et al.* [5]. Such decision processes also generalize MMDPs to continuous-time with asynchronous events. We use a restricted GSMDP and hence do not present that framework thoroughly.

The state variables for this problem describe the position of each agent and the idleness of each vertex (as per equation (1)). If the total number of agents is $N$, the state space is

$$\mathcal{S} = V^N \times [0,1]^{|V|}\,.$$

Given some state $s = (\mathbf{v}, \mathbf{k}) \in \mathcal{S}$, $v_i$ is the position of the $i$-th agent and $k_i$ the idleness of the $i$-th vertex. We use $s^t = (\mathbf{v}^t, \mathbf{k}^t)$ for the state and its components at time $t$.

At various time points, called decision epochs, the agents must choose an action. The actions from which an agent can choose from depend on the structure of the graph and on its position: if an agent is at vertex $v$, it can choose its action from $\mathcal{A}_v = \{u : [v,u] \in E\}$. If an agent chooses action $u$ from vertex $v$ at time $t^i$, the next decision epoch for that agent occurs at time $t^{i+1} = t^i + L_{vu}$, and $v^t = v$ while $t \in [t^i, t^{i+1})$ and $v^t = u$ as soon as $t = t^{i+1}$.

The problem is concurrent because the decision epochs of all agents can be interleaved arbitrarily. Each component $k_i$ of $\mathbf{k}$ evolves independently. Equation (1) was defined in terms of two time epochs ($t$ and $t + \Delta t$) and the number of agents ($n$). Let $\{t^j\}_j$ be the non-decreasing sequence of decision epochs and write $n_i^j$ for the number of agents arriving at vertex $i$ at time $t^j$. To simplify notation, let $\Delta t^j = t^{j+1} - t^j$. We thus have that

$$k_i^{t^{j+1}} = k_i^{t^j} a^{n_i^{j+1}} b^{\Delta t^j} + 1 - a^{n_i^{j+1}}\,.$$

The reward process $\mathcal{R}$ is defined in terms of $\mathbf{k}$. Specifically, the rate at which reward is gained is given by

$$dR = \mathbf{w}^\top \mathbf{k}^t \, dt. \tag{2}$$

The discounted value function for a GSMDP is defined by Younes *et al.* [5]. In our problem, it becomes:

$$\begin{aligned} V^\pi(s) \underset{(a)}{=} \mathbb{E}\left[\int_0^\infty \gamma^t dR\right] &\underset{(b)}{=} \mathbb{E}\left[\sum_{j=0}^\infty \gamma^{t^j} \int_0^{\Delta t^j} \gamma^t \mathbf{w}^\top \mathbf{k}^t \, dt\right] \\ &\underset{(c)}{=} \mathbb{E}\left[\sum_{j=0}^\infty \gamma^{t^j} \mathbf{w}^\top \mathbf{k}^{t^j} \frac{(b\gamma)^{\Delta t^j} - 1}{\ln(b\gamma)}\right], \end{aligned} \tag{3}$$

where $\gamma \in (0,1]$ is the discount factor. Equality (a) is the definition of the continuous-time discounted value function, (b) is obtained by noticing that (2) only exists between decision epochs and (c) is obtained by evaluating the integral. The expectation is taken over action durations. In this paper we consider deterministic durations. The sequence of states and decision epochs encountered depends on the actions chosen by the agents, which are denoted by $\pi$. The problem is to chose actions for all agents and decision epochs that will maximize this expectation.

# 4 Solving the Patrolling Problem

Online planning has the advantage that it solves (3) only for the current state. This is in contrast with offline algorithms that do so for all states. Online algorithms are more appealing for systems that need to make a decision for the situation at hand. The problem described in section 3 is simpler to solve online than offline. We use an anytime online planning algorithm, which is described in section 4.1. We describe how it can be applied to the patrol problem in sections 4.2 and 4.3. An algorithm to coordinate multiple agents while retaining the online and anytime properties in section 4.4. An algorithm is anytime if firstly it can be stopped at any time and provide a useful result and secondly running it any longer does not degrade solution quality.

## 4.1 Anytime Error Minimization Search

Anytime Error Minimization Search (AEMS) is an online algorithm introduced originally by Ross *et al.* [6] for Partially Observable Markov Decision Processes (POMDPs). It performs a heuristic search in the state space. The search proceeds using a typical branch and bound scheme. Since the exact long term expected value of any state is not exactly known, it is approximated using upper and lower bounds. AEMS guides the expansion of the search tree by greedily reducing the error on the estimated value of the root node. While we are not in a strict POMDP setting, the greedy error reduction is useful. In our problem, actions have the same interpretation as in a partially observable setting, whereas observations are the travel durations. Using many such "observations", our model is extendable to stochastic durations. Let us recall briefly how AEMS works.

In AEMS, the error is defined using the upper bound and the lower bound on the value of some state. Let $s \in \mathcal{S}$ be a state. We have $L(s) \leq V(s) \leq U(s)$ where $V(s)$ is the actual value of $s$, and $L(s)$ and $U(s)$ are the lower and upper bounds respectively. Given some search tree $\mathbb{T}$, whose set of leaf nodes is noted $\mathcal{F}(\mathbb{T})$, the bounds for the root node are estimated recursively according to

$$L(s) = \begin{cases} \hat{L}(s) & \text{if } s \in \mathcal{F}(\mathbb{T}) \\ L(s, \boldsymbol{a}) = \max_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} R(s, \boldsymbol{a}) + \gamma L(\tau(s, \boldsymbol{a})) & \text{otherwise.} \end{cases} \tag{4}$$

and

$$U(s) = \begin{cases} \hat{U}(s) & \text{if } s \in \mathcal{F}(\mathbb{T}) \\ U(s, a) = \max_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} R(s, \boldsymbol{a}) + \gamma U(\tau(s, \boldsymbol{a})) & \text{otherwise,} \end{cases} \tag{5}$$

where $\tau(s, \mathbf{a})$ is the next state if action $\mathbf{a}$ is taken in state $s$. In equations (4) and (5), $\hat{L}(s)$ and $\hat{U}(s)$ are problem-dependent heuristics such as those proposed in section 4.2.

An estimation of the error on the value of $s$ is given by $\hat{e}(s) = U(s) - L(s)$. Let $s^0$ be the state at the root of search tree $\mathbb{T}$. We are interested in expanding the state at the fringe of the search tree whose contribution to the error on $s^0$ is maximal. Since all states are not reachable with equal probability (depending on the policy), the contribution of any state $s$ to the error on $s^0$ is approximated by:

$$\hat{E}(s^0, s^t, \mathbb{T}) = \gamma^t \Pr(h_0^t | s^0, \hat{\pi}) \hat{e}(s^t),$$

where $t$ is the depth of $s$ in $\mathbb{T}$, and $\Pr(h_0^t|s^0, \hat{\pi})$ denotes the probability of having history $h_0^t$ (the sequence of joint-actions that lead from $s^0$ to $s^t$), while following policy $\hat{\pi}$. Note that the above term describes exactly the error whenever $\hat{\pi} = \pi^*$. The value of $\Pr(h_0^t|s^0, \hat{\pi})$ is what we are now interested in.

If $h_0^t = \boldsymbol{a}^0, \boldsymbol{o}^0, \boldsymbol{a}^1, \boldsymbol{o}^1, \ldots, \boldsymbol{a}^t, \boldsymbol{o}^t$, is the joint-action history for some sequence of states $s^0, s^1, \ldots s^t$, then we have

$$\Pr(h_0^t|s^0, \hat{\pi}) = \prod_{i=0}^{t} \Pr(\boldsymbol{a}^i = \hat{\pi}(s^i)|s^i) \Pr(\boldsymbol{o}^i|s^i, \boldsymbol{a}^i).$$

Since we do not know the optimal policy (this is what we are searching for), a good approximation is to use:

$$\Pr(\boldsymbol{a}|s) = \begin{cases} 1 & \text{if } U(s, \boldsymbol{a}) = \max_{\boldsymbol{a}' \in \boldsymbol{\mathcal{A}}} U(s, \boldsymbol{a}') \\ 0 & \text{otherwise.} \end{cases}$$

Given a search tree $\mathbb{T}$, rooted at $s^0$, AEMS tells us that the next state to expand is

$$\tilde{s}(\mathbb{T}) = \operatorname*{arg\,max}_{s \in \mathcal{F}(\mathbb{T})} \hat{E}(s, s^0, \mathbb{T}).$$

Each time a node $\tilde{s}$ is expanded, it is removed from $\mathcal{F}(\mathbb{T})$, its children are added to $\mathcal{F}(\mathbb{T})$ and the bounds of $\tilde{s}$ and its parents are updated. When an agent must choose an action, the action of maximum lower bound is chosen.

### 4.2 Bounds for the Patrolling Problem

In order to use AEMS, we must specify the lower ($\hat{L}(\cdot)$) and upper ($\hat{U}(\cdot)$) bounds for the value of states. The alternate representation of idleness introduced in section 2 makes the reward and value functions bounded. It is thus possible to provide upper and lower bounds for the value function.

A lower bound for the value of any state is the value of following any policy from that state. A greedy policy is arguably a good “simple” policy. It is defined to always choose the action with arrival state for which $\mathbf{w}^\top \mathbf{k}$ is maximal. Equation (3) defines the value of such a policy.

An upper bound is usually obtained by relaxing problem constraints. We can thus upper-bound the value of a policy by assuming that agents are ubiquitous: they can be in more than one locations at the same time. Whenever an agent reaches a vertex, it instantaneously multiplies itself and starts heading to adjacent unvisited locations. This bound estimates the shortest time that a swarm of agents would take to cover the entire graph and estimates through the discount factor an upper bound on the maximum reward obtainable. This bound implicitly assumes that the optimal policy does not require having more than one agent at any vertex.

### 4.3 Extension to Asynchronous Multiagent Setting

Extending AEMS to a asynchronous multiagent is simple: whenever a node is expanded, there is a branch for every joint action (and observation). Asynchronicity is handled with state augmentation. The state is now $(s, \boldsymbol{\eta})$, where $\eta_i$ is the time remaining before the next decision epoch of agent $i$. At any time $t$, the next decision epoch happens at time $t + \min_i \{\eta_i\}$. The expand operation adds actions and observations for any agent for which $\eta = 0$. Whenever agent $i$ performs an action of duration $\Delta t$, $\eta_i$ is assigned $\Delta t$. Otherwise, $\eta_i$ is updated according to its depth in the search tree.

### 4.4 Coordinating Agents

AEMS can be used to perform online planning for any subset of agents. However, it is unlikely that any agent has the computational capacity to compute a joint policy, because the complexity is exponential in the number of agents. We thus coordinate agents *locally*. We define an partial order amongst agents. (Think of a directed acyclic graph where agents are vertices and there is an edge between two agents whenever they are close.) We say that an agent is greater than (resp. less than) another agent if it must choose its policy before (resp. after).

The agents compute their policy according to that order. Once an agent knows the policies of all greater agents, it proceeds to compute its policy, and then communicates it to the lesser agents. Whenever an agent selects its policy, it chooses the best policy given the policy of greater agents. This approach can be improved by using *altruistic* agents. Loosely speaking, such an agent does not consider only the value of its own policy. Instead it chooses the best sub-policy from a *small* set of joint policies: i.e. the joint policies for himself and a small number (i.e. 1) of lesser neighboring agents.

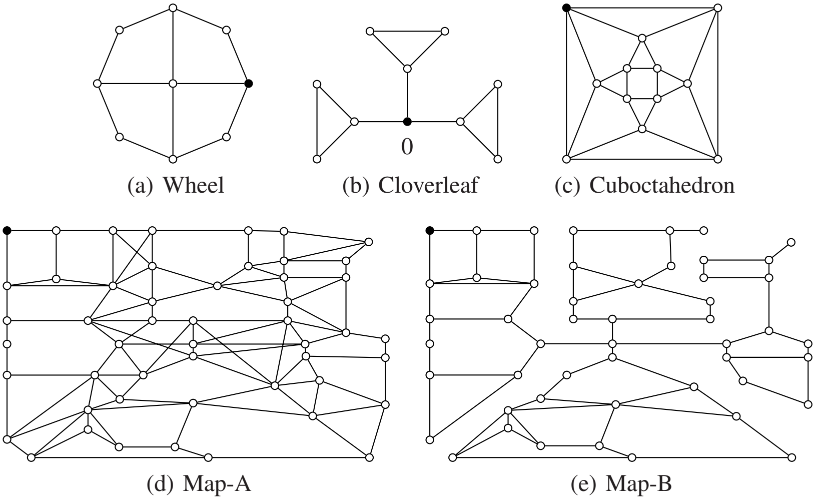
A useful property of this coordination algorithm is that if the agents use an online anytime planner, then it is also anytime and online. A fallback strategy is to ignore the presence of the greater agents until their policy has been received. For the remainder of this paper, we use C-AEMS to refer to this coordinated AEMS.

## 5 Experiments

Comparisons are made against a reactive algorithm proposed by Machado *et al.* [4] since it handles uncertainties and weights on locations. This algorithm is actually equivalent to the lower bound we used. We also compare our algorithm to the optimal policy value for some specific graphs that have an Hamiltonian cycle.

While our algorithm maximizes expected reward, idleness is often used a performance measure for patrol planning algorithms. Idleness is a useful measure because it is more readily interpretable than the average reward rate. The instantaneous idleness is the maximum idleness among all vertices at some given time. The maximum idleness is the maximum instantaneous idleness throughout an episode whereas the mean idleness is the mean instantaneous idleness. For the idleness measure to be meaningful, we have chosen $\mathbf{w} = \mathbf{1}$ and $a = 0$. (It is easier to verify that a patrol route is optimal in that case.) We have used $\gamma = 0.95$ throughout. The choice of $b$ is discussed in section 6. The initial state is $\mathbf{k} = \mathbf{1}$ for all experiments and the start vertex is shown as filled circles in Figure 1.

Results presented in Table 1 show the performance of our algorithm, for two agents, on various graphs instances presented in Figure 1. The small problem instances were



(a) Wheel (b) Cloverleaf (c) Cuboctahedron

(d) Map-A (e) Map-B

**Fig. 1.** Patrolling instances. Start vertex is filled, edges have unit length and vertices have unit weight unless otherwise noted.

**Table 1.** Results for 2 agents on different graphs

| | $b$ | Max Idleness | Mean Idleness | Total Reward |
|---|---|---|---|---|
| Wheel (50 time units) | | | | |
| c-AEMS | 0.9 | **5** | **3.90** | **144.512** |
| Greedy | | 6 | 4.24 | 142.957 |
| Clover (50 time units) | | | | |
| c-AEMS | 0.9 | **8** | **5.76** | **153.255** |
| Greedy | | 11 | 6.19 | 150.364 |
| Cuboctahedron (50 time units) | | | | |
| c-AEMS | 0.9 | **5** | **4.29** | **183.317** |
| Greedy | | 6 | 4.33 | 182.873 |
| Map-A (200 time units) | | | | |
| c-AEMS | 0.95 | **37** | **26.34** | **5718.02** |
| Greedy | | 51 | 32.58 | 5567.42 |
| Map-B (200 time units) | | | | |
| c-AEMS | 0.99 | **59** | **38.62** | **8708.29** |
| Greedy | | 67 | 41.34 | 8477.87 |

chosen for their simplicity and also because the optimal policy is somewhat obvious. The wheel and cuboctahedron instances are interesting because of their symmetry. The two large instances, Map-A and Map-B, are seen frequently in the literature, for instance in Santana *et al.* [1]. The AEMS algorithm was allowed to expand 50 states on the three small instances and 100 on the larger ones.

## 6 Discussion

The parameters of the model, ($\gamma$, $\mathbf{w}$, $a$ and $b$) influence greatly the policy found by our algorithm. If $b$ is too small, locations that have not been visited for a long time can remain unvisited without significant value loss. In such cases, the route of maximum value does not correspond to a low idleness policy. In practice, the representation of $\mathbf{k}$ is also subject to underflow if $b$ is too small. If there is a circuit of length $L$, a value of $b$ that would incite agents to visit all vertices is one such that $0.5 \approx b^L$. Note that while the policy generated by the reactive algorithm does not depend on $a$ or $b$, the rewards it receives depends on $b$. Since $b$ represents the rate at which the vertices change, it is typically not a tunable parameter. Interesting problems about $b$ include finding how many agents are required to attain a given performance level and what is $b$ given an actual problem instance. Although we did not explicitly state it in the definition of the model, $b$ could be different from one vertex to another and $a$ could be different for every agent-vertex pair.

Results suggest that for given instances, and some values of $b$, the model allows finding good solutions with regard to the max-idleness metric. This is not obvious, because our reward is based on the exponential idleness of section 2. Our approach slightly outperforms the baseline approach on the instances, whereas the improvement is more significant on the large ones. We attribute this to non-myopia and to more efficient coordination between agents. On the three small instances, our algorithm found the optimal patrol route.

A problem with C-AEMS is that it does not perform well on highly symmetric instances such as the wheel or the cuboctahedron. In such cases, lot of computational effort is spent in discriminating patrol routes that have exactly the same value which is not possible. AEMS will then behave like breadth-first search. Such problems happen especially at the beginning of a simulation, because almost all vertices have never been visited and have equal value.

Our C-AEMS spends a significant portion of its time computing the value of the lower bound. The algorithm performs better when the bound is computed accurately. However, to get a lower error, the value must be evaluated over a longer time span. Computing the value of the lower bound offline would alleviate this problem.

## 7 Conclusion

We defined a model for the stochastic multiagent patrolling problem using Markov models. The proposed model allows specifying time uncertainty and concurrent asynchronous actions by the agents. We propose an online algorithm to solve the problem for a subset of agents and a method for coordinating many such solvers.

The algorithm can be implemented on physical robots that must perform patrol under the assumption that they can communicate each other's policies reliably. The robots must be provided with a way to estimate the time until they reach the patrol location they are currently heading to. The proposed framework supports distributions on travel durations. Future work includes improving the performance of the lower bound with supervised learning and performing experiments in a setting where the agents evolve in a simulated world and to eventually allow actual UAVs to patrol a set of locations.

## References

1. Santana, H., Ramalho, G., Corruble, V., Ratitch, B.: Multi-agent patrolling with reinforcement learning. In: Proc. of AAMAS 2004 (2004)
2. Glad, A., Simonin, O., Buffet, O., Charpillet, F.: Theoretical study of antbased algorithms for multi-agent patrolling. In: Proc. of ECAI 2008, pp. 626–630 (2008)
3. Chevaleyre, Y., Sempé, F., Ramalho, G.: A theoretical analysis of multi-agent patrolling strategies. In: Proc. of AAMAS 2004 (2004)
4. Machado, A., Ramalho, G., Zucker, J.D., Drogoul, A.: Multi-Agent Patrolling: an Empirical Analysis of Alternative Architectures. In: Proc. of the Multi-Agent Based Simulation Conference (2002)
5. Younes, H.L.S., Simmons, R.G.: A formalism for stochastic decision processes with asynchronous events. In: Proc. of AAAI Workshop on Learning and Planning in Markov Processes, pp. 107–110. AAAI Press, Menlo Park (2004)
6. Ross, S., Chaib-draa, B.: AEMS: An Anytime Online Search Algorithm for Approximate Policy Refinement in Large POMDPs. In: Proc. of IJCAI 2007, pp. 2592–2598 (2007)

# Hybrid Framework to Image Segmentation

Fernando C. Monteiro

Polytechnic Institute of Bragança, Portugal
monteiro@ipb.pt

**Abstract.** This paper proposes a new hybrid framework to image segmentation which combines edge- and region-based information with spectral techniques through the morphological algorithm of watersheds. The image is represented by a weighted undirected graph, whose nodes correspond to over-segmented regions (atomic regions), instead of pixels, that decreases the complexity of the overall algorithm. In addition, the link weights between the nodes are calculated through the intensity similarities combined with the intervening contours information among atomic regions. We outline a procedure for algorithm evaluation through the comparison with some of the most popular segmentation algorithms: the mean-shift-based algorithm, a multiscale graph based segmentation method, and JSEG method for multiscale segmentation of colour and texture. Experiments on the Berkeley segmentation database indicate that the proposed segmentation framework yields better segmentation results due to its region-based representation.

## 1 Introduction

Image segmentation is an important component in many image understanding algorithms and practical vision systems, and aims at identifying regions that have a specific meaning within images. Another definition of image segmentation is the identification of regions that are uniform with respect to some parameter, such as image intensity or texture. While the latter definition is often used for technical reasons, the former definition should be preferred from an application point of view. Although the effort made in the computer vision community there is no algorithm that is known to be optimum in image segmentation. Much research is being done to discover new methods building up on previous ideas [1, 5, 7].

The main goal of this paper is to develop an algorithm for efficient segmentation of a grey level image that a) identifies perceptually homogeneous regions in the images, b) makes minimal assumptions about the scene, and c) avoids merging of multiple objects into single segments and vice-versa. The definition of a new structure for region-based graph, the presentation of a new similarity function, and the application of multiclass normalized cuts to group atomic regions are the main contributions of this paper.

The combination of watershed and spectral methods solves the weaknesses of each method by using the watershed to provide small prototype regions from which similarity graph can be obtained. Rather than clustering single feature

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 657–666, 2009.

points we will cluster micro-regions, confident that the underlying primitive regions are reliable. Our approach actually prefers the objects to be over-segmented into a number of smaller regions to ensure that a minimal amount of background is connected to any of the object regions.

The algorithm described in this paper can be well classified into the category of hybrid techniques, since it combines the edge-based, region-based and the morphological techniques together through the spectral-based clustering approach. Rather than considering our method as another segmentation algorithm, we propose that it can be considered as an image segmentation framework within which existing image segmentation algorithms that produce over-segmentation may be used in the preliminary segmentation step.

This paper is organized as follows: in Section 2 the overview of the multiclass spectral segmentation algorithm is given. The details of the proposed framework are explained in Section 3. After presenting the experimental results and evaluation in Section 4, the final section is devoted to the concluding remarks.

## 2 Overview of Multiclass Normalized Cuts

Spectral approach is a global graph-based segmentation method that uses the eigenvectors and eigenvalues of a matrix derived from the pairwise similarities of graph nodes. The problem of image segmentation based on pairwise similarities can be formulated as a graph partitioning problem in the following way: consider the weighted undirected graph $G = (V, E, W)$ where each node $v_i \in V$ corresponds to a locally extracted image features, e.g. pixels and the links in $E$ connect pairs of nodes. A weight $w_{i,j} \in \mathbb{R}_0^+$ is associated with each link based on some property of the nodes that it connects (e.g., the difference in intensity, colour, motion, location or some other local attribute).

Let $\Gamma = \{V_i\}_{i=1}^k$ be a multiclass disjoint partition of $V$ such as $V = \cup_{i=1}^k V_i$ and $V_i \cap V_j = \emptyset, i \neq j$. Image segmentation is reduced to the problem of partitioning the set $V$ into disjoint non-empty sets of nodes $(V_1, .., V_k)$, such similarity among nodes in $V_i$ is high and similarity across $V_i$ and $V_j$ is low. The solution in measuring the goodness of the image partitioning is the minimization of the normalized cut as a generalized eigenvalue problem.

Shi and Malik [8] introduced the normalized cut $(NCut)$ criterion for bipartitioning segmentation. Let $V_A, V_B$ be two disjoint sets of the graph $V_A \cap V_B = \emptyset$. We define $links\,(V_A, V_B)$ to be the total weighted connections from $V_A$ to $V_B$:

$$links\,(V_A, V_B) = \sum_{i \in V_A, j \in V_B} w_{i,j} \ . \tag{1}$$

The intuition behind the normalized cut criterion is that not only we want a partition with small link cut but we also want the subgraphs formed between the matched nodes to be as dense as possible. This latter requirement is partially satisfied by introducing the normalizing denominators in the Eq. (1). The normalized cut criterion for a bipartition of the graph is then defined as follows:

$$Ncut\left(A,B\right)=\frac{links\left(A,B\right)}{links\left(A,V\right)}+\frac{links\left(A,B\right)}{links\left(B,V\right)}\ . \tag{2}$$

By minimizing this criterion we simultaneously minimize the similarity across partitions and maximize the similarity within partitions. This formulation allows us to decompose the problem into a sum of individual terms and formulates a dynamic programming solution finding a partition $\Gamma$ that minimizes the multiclass normalized cut $(kNCut)$

$$kNCut\left(\Gamma\right)=\frac{links\left(V_1,\overline{V_1}\right)}{links\left(V_1,V\right)}+\frac{links\left(V_2,\overline{V_2}\right)}{links\left(V_2,V\right)}+...+\frac{links\left(V_k,\overline{V_k}\right)}{links\left(V_k,V\right)}\ , \tag{3}$$

where $\overline{V_i}$ represents the complement of $V_i$.

For a fixed $k$ partitioning of the nodes of $G$, reorder the rows and columns of $W$ accordingly so that the rows of $W$ correspond to the nodes in $V_i$. Let $D=diag\left(D_1,...,D_k\right)$ be the $n\times n$ diagonal matrix so that $D_i$ is given by the sum of the weights of all links on node $i$: $D_i=\sum_{j=1}^{k}W_{ij}$. It is easy to verify that

$$links\left(V_i,\overline{V_i}\right)=D_i-W_{ii}\qquad\text{and}\qquad links\left(V_i,V\right)=D_i\ . \tag{4}$$

A multiclass partition of the nodes of $G$ is represented by an $n\times k$ indicator matrix $\mathbf{X}=\left[\mathbf{x_1},...,\mathbf{x_k}\right]$ where $X\left(i,l\right)=1$ if $i\in V_l$ and 0 otherwise [9]. Since a node is assigned to one and only one partition there is an exclusion constraint between columns of $X$: $XI_k=I_n$. It follows that

$$links\left(V_i,\overline{V_i}\right)=\mathbf{x_i}^T\left(D-W\right)\mathbf{x_i}\qquad\text{and}\qquad links\left(V_i,V\right)=\mathbf{x_i}^TD\mathbf{x_i}\ . \tag{5}$$

Therefore,

$$\begin{aligned}kNCut\left(\Gamma\right)&=\frac{\mathbf{x_1}^T\left(D-W\right)\mathbf{x_1}}{\mathbf{x_1}^TD\mathbf{x_1}}+...+\frac{\mathbf{x_k}^T\left(D-W\right)\mathbf{x_k}}{\mathbf{x_k}^TD\mathbf{x_k}}\\&=k-\left(\frac{\mathbf{x_1}^TW\mathbf{x_1}}{\mathbf{x_1}^TD\mathbf{x_1}}+...+\frac{\mathbf{x_k}^TW\mathbf{x_k}}{\mathbf{x_k}^TD\mathbf{x_k}}\right)\ ,\end{aligned} \tag{6}$$

subject to $X^TDX=I_k$.

The solution for the generalized Rayleigh quotients that compose Eq. (6) is the set of eigenvectors $X$ associated with the set of the smallest eigenvalues $\Phi=\left\{0=\nu_1\leq...\leq\nu_k\right\}$ of the system

$$\left(D-W\right)X=\Phi DX\ . \tag{7}$$

However, this problem is NP-hard [8] and therefore generally intractable. If we ignore the fact that the elements of $\mathbf{x_i}$ are either zero or one, and allow them to take continuous values, by using the method of Lagrange multipliers, Eq. (7) can be expressed by the standard eigenvalue problem. Let $\mathbf{y_i} = D^{1/2}\mathbf{x_i}$ and $Y = [\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_k}]$.

$$\widetilde{W}Y = Y\Lambda \ , \tag{8}$$

subject to $Y^TY = I_k$, where $\widetilde{W} = D^{-1/2}WD^{-1/2}$ is the normalized graph Laplacian matrix[1], with $\Lambda = \{1 = \lambda_1 \geq ... \geq \lambda_k\}$ where $\lambda_i = 1 - \nu_i$.

If $Y$ is formed with any $k$ eigenvectors of $\widetilde{W}$ then $\Lambda$ is the $k \times k$ diagonal matrix formed with the eigenvalues corresponding to the $k$ eigenvectors in $Y$. These $k$ eigenvectors must be distinct to satisfy $Y^TY = I_k$. This means that

$$Y^T\widetilde{W}Y = Y^TY\Lambda = I_k\Lambda = \Lambda \ , \tag{9}$$

and the *trace* of $Y^T\widetilde{W}Y$ is the sum of the eigenvalues corresponding to the $k$ eigenvectors in $Y$. So, Eq. (6) becomes equivalent to

$$kNCut\,(\Gamma) = k - trace\left(Y^T\widetilde{W}Y\right) = k - \sum_{i=1}^{k}\lambda_i \ . \tag{10}$$

It follows from Fan's Theorem that the maximum on the right hand side of Eq. (10) is achieved when $Y$ is taken to by any orthonormal basis for the subspace spanned by the eigenvectors corresponding to the $k$ largest eigenvalues of $\widetilde{W}$. From this we establish a lower bound $l\,(k)$ on $kNCut\,(\Gamma)$ as

$$\min_{\Gamma} NCut_k\,(\Gamma) \geq k - \sum_{i=1}^{k}\lambda_i \ , \tag{11}$$

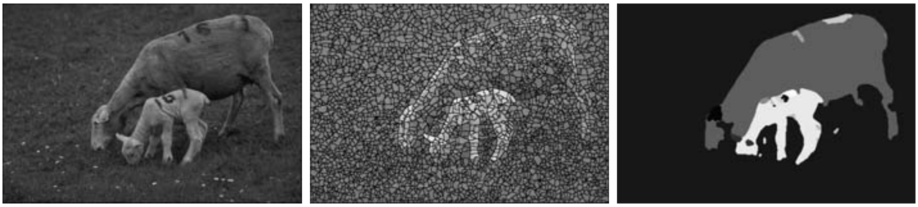
where $\lambda_1, ..., \lambda_k$ are the $k$ largest eigenvalues of $\widetilde{W}$.

For $k = 2$ the bound becomes $l\,(2) = 2 - (1 + \lambda_2) = 1 - \lambda_2 = \nu_2$ that is the second smallest eigenvalue of the generalized eigensystem of Eq. (7). This is consistent with the bi-partitioning method proposed by Shi and Malik [8].

## 3 Proposed Hybrid Framework

The proposed methodology has three major stages (see Fig. 1). First, we reduce image noise, as a pre-processing stage, using an anisotropic filter, and create an over-segmented image based on the watershed transform of the gradient image. In the second stage, the over-segmented image will be the input for the region similarity graph (RSG) construction. Finally, we apply a spectral approach on the RSG. This framework integrates edges and region-based segmentation with spectral-based clustering through the morphological watersheds.

[1] Although the Laplacian matrix is usually represented by $I - \widetilde{W}$, replacing $\widetilde{W}$ with $I - \widetilde{W}$ only changes the eigenvalues (from $\lambda$ to $1 - \lambda$) and not the eigenvectors.



(a) Input image (b) Atomic regions (c) Segmentation

**Fig. 1.** Example of image segmentation. (a) Input image. (b) Atomic regions. Each atomic region is a node in the graph $G$. (c) Segmentation (labelling) result.

### 3.1 Watershed Over-Segmentation

A critical issue in watershed techniques is known to be over-segmentation i.e. the tendency to produce too many basins [1]. Several methods have been proposed in the literature to reduce the spurious boundaries created due to noise and produce a meaningful segmentation.

In this work we provide three methods to overcome this problem. First, bilateral anisotropic filtering can be applied to remove noise from the image. Secondly, some of the weakest edges are removed by a gradient minima suppression process known as *pre-flooding*. This concept uses a measure of depth of a certain basin. Prior to the transform, each catchment basin is flooded up to a certain height above its bottom, i.e. the lowest gradient magnitude and it can be thought as a flooding of the topographic image at a certain level (flooding level). This process will create a number of lakes, grouping all the pixels that lie below the flooding level.

The third one, handles to control over-segmentation eliminating spurious tiny regions associated with uniform regions through a merging step. This eliminates tiny regions which have similar adjacent regions, while maintaining the accuracy of the partition. This stage is required to reduce the computational complexity in the graph partitioning. Another advantage of these steps is to prevent large homogeneous (flat) regions from being split in the graph-based segmentation (a common problem with balanced graph cut methods).

### 3.2 Region Similarity Graph

Spectral-based methods use the eigenvectors and eigenvalues of a matrix derived from the pairwise similarities of features (pixels or regions). This effect is achieved by constructing a fully connected graph. Considering all pairwise pixel relations in an image may be too computational expensive. Unlike other well known clustering methods [8, 9] which use down-sampling pixel-based to construct the graph, our method is based on selecting links from a weighted undirected graph $G = (V, E, W)$ based on a region similarity graph where each node corresponds to an atomic region.

Some characteristics of the RSG model that yield to relevant advantages with regard to the region adjacency graph model are: i) the RSG allows the existence of links between pairs of non-adjacent regions. ii) it is defined once and it does not need any dynamic updating when merging regions.

The proposed RSG structure takes advantage on region-based representation. The set of nodes $V$ are represented by the centroid of each micro-region. The sets of links $E$ and link weights $W$ represent, respectively, relationships and similarity measures between pair of regions.

**Pairwise Spatial Similarity.** Each region $r_i$ represents a small group of pixels where the centroid $\overline{x}_i$ is utilized as a node of the graph. For each pair of nodes, node similarity is inversely correlated with the maximum contour energy encountered along the line connecting the centroids of the regions. If there are strong contours along a line connecting two centroids, these atomic regions probably belong to different segments and should be labelled as dissimilar. Let $i$ and $j$ be two atomic regions and the orientation energy $OE^*$ between them, then the intervening contours contribution to the link weight is given by:

$$w_{ic}\left(i,j\right) = \exp\left[-\frac{\max_{t\in line(i,j)} \left\|OE^*\left(\overline{x}_i,\overline{x}_j\right)\right\|^2}{\sigma_{ic}^2}\right] \quad , \tag{12}$$

where $line\left(i,j\right)$ is the line between centroids $\overline{x}_i$ and $\overline{x}_j$ formed by $t$ pixels.

The mean intensity of each node contributes for the link weight according to the following function:

$$w_I\left(i,j\right) = \exp\left(-\frac{\left(I_{\overline{x}_i} - I_{\overline{x}_j}\right)^2}{\sigma_I^2}\right) \quad . \tag{13}$$

These cues are combined in a final link weight similarity function, with the values $\sigma_{ic}$ and $\sigma_I$ selected in order to maximize the dynamic range of $\mathbf{W}$:

$$\mathbf{W}\left(i,j\right) = w_{ic}\left(i,j\right) \cdot w_I\left(i,j\right) \quad . \tag{14}$$

In almost all the graph-based approaches proposed in the literature the spatial distance cue is also used to compute the similarity between graph nodes. However, during our experiments, we note that such cue is responsible for the partition of homogeneous areas in the image - an issue commonly associated to normalized cut algorithm. Instead, we use intervening contours which are equivalent to spatial distance without suffering from the same problems [7].

**Implementation Details of the RSG.** For a computational consideration it is important to sort and label all the regions created by the watershed segmentation. In the following some implementation details are given about the construction of the RSG. For each region $r_i$, spatial location $\overline{x}_i$ is computed as centroids of their pixels. Two dynamic data structures are used through which it is very convenient to add or remove regions: 1) A label map in which each pixel value corresponds to the label of the segment that this pixel belongs to; 2) An array of segments where each segment is represented by a linked-list of pixels which correspond to the pixels that belong to the segment. This list includes the location and the grey-level of each pixel.

This dual representation of a partitioned image allows for a very efficient implementation. The label map grants us immediate access to the label of every pixel in the image. The array of lists gives us immediate access to the set of pixels that belong to each segment. Using this representation two different segments can be merged into one by iterating through the corresponding linked-lists and updating the label map. Even more, we can easily obtain the centroid and the mean value of each segment.

To compute the similarity matrix the current approach uses only image brightness and magnitude gradient. Additional features such as texture, could be added to the similarity criterion. This may slow the construction of the RSG but the rest of the algorithm will proceed with no change.

## 4 Experiments and Evaluation

In order to evaluate the performance of the proposed method experiments have been conducted to compare it with some of the most popular algorithms: (i) mean shift (EDISON) [2], (ii) a multiscale graph based segmentation method (MNCUT) [3], and (iii) JSEG [4]. To provide a numerical evaluation measure and thus allow comparisons, the experiments for the evaluation were conducted on the manual segmentations of the publicly available Berkeley Segmentation Dataset[2] [6]. Due to the large number of images, we choose to apply our framework to images with horses on it. The task is cast as a boundary detection problem, with results presented in terms of Precision (P) and Recall (R) measures. The algorithm provides a binary boundary map which is scored against each one of the hand-segmented results of Berkeley Dataset, producing a $(R, P, F)$ value. The final score is given by the average of those comparisons. The quantitative evaluation of segmentation results (F-measure) is summarized in Table 1.

The proposed approach produces segmentations of high quality and with better results than the other methods for all tested images. This new approach overcomes some limitations usually associated with spectral clustering approaches. Some segmentation results of the proposed method are shown in Fig. 2. Since the F-measure is a boundary-based measure the segmentation results are presented as boundaries over the original images.

### 4.1 Robustness to Noise

To analyse the behaviour of the algorithm in presence of noise, the images were corrupted with four levels of Gaussian additive noise with standard deviations $\sigma = 5, 10, 20, 30$. The effect of the pre-processing step in reducing the noise, with

[2] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

**Table 1.** Results of quantitative evaluation in terms of F-measure for the comparison between the proposed method (WNCUT), Mean shift (EDISON), JSEG and the multiscale segmentation MNCUT

| Method | 28075 | 113016 | 113044 | 197017 | 216041 | 220075 | 291000 | 361010 |
|---|---|---|---|---|---|---|---|---|
| WNCUT | 0.66 | 0.64 | 0.66 | 0.72 | 0.65 | 0.65 | 0.69 | 0.74 |
| EDISON | 0.61 | 0.41 | 0.47 | 0.69 | 0.60 | 0.36 | 0.35 | 0.57 |
| JSEG | 0.42 | 0.55 | 0.53 | 0.46 | 0.40 | 0.32 | 0.41 | 0.49 |
| MNCUT | 0.25 | 0.30 | 0.35 | 0.48 | 0.36 | 0.37 | 0.24 | 0.58 |

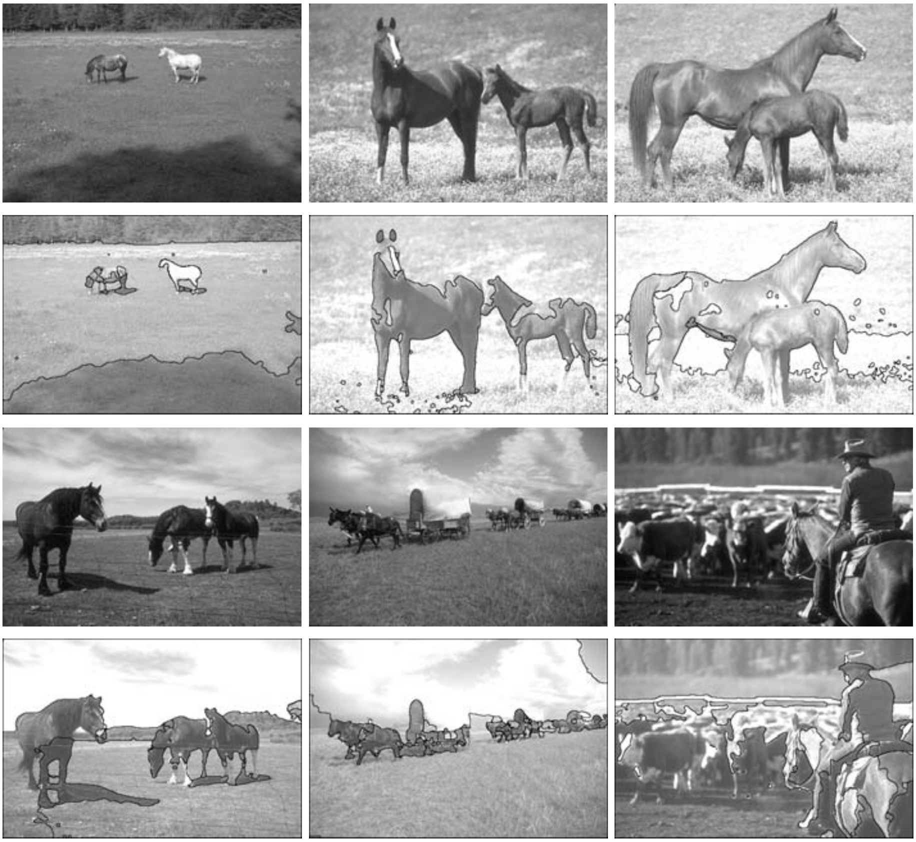


**Fig. 2.** Segmentation results: First and third rows: original images (first six images of Table 1). Second and fourth rows: segmentation results with the proposed method.

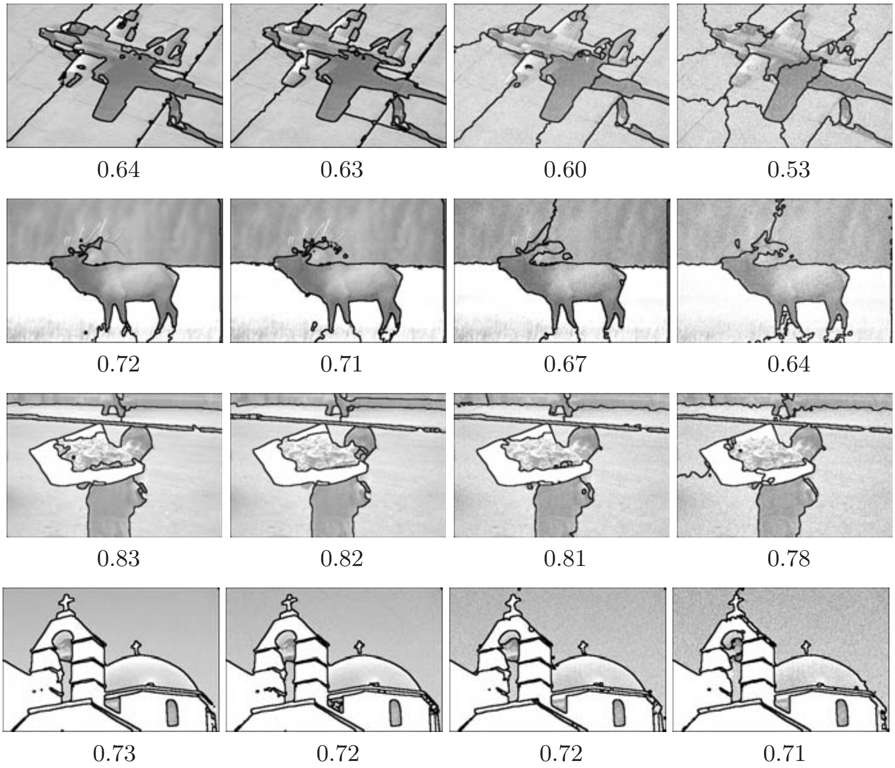


**Fig. 3.** Performance of the proposed approach on noisy images. Results with added Gaussian noise with $\sigma$, from left to right, equal to 5, 10, 20, 30. The values below the images are the F-measures.

a reduction on the number of irrelevant regions in the output of the watershed algorithm, can be observed in Fig. 3.

Our method turned out to be extremely robust to artificially added Gaussian noise. We may notice that segmentation results are not very affected till $\sigma = 20$, and it produces a good segmentation even for added Gaussian noise with an amplitude of $\sigma = 30$. This amount of noise is greater than would be expected in a normal real image.

## 5 Conclusion

This paper proposes an image segmentation approach which combines edge- and region-based information with spectral techniques through the morphological algorithm of watersheds. Using small atomic regions instead of pixels leads to a more natural image representation - the pixels are merely the result of the digital image discretization process and they do not occur in the real world. Besides producing smoother segmentations than pixel-based partitioning methods, it also reduces the computational cost in several orders of magnitude. The experimental results demonstrate the effectiveness of the proposed approach to compare favourably with some of the most popular image segmentation methods.

## References

[1] O' Callaghan, R.J., Bull, D.R.: Combined morphological-spectral unsupervised image segmentation. IEEE Transactions on Image Processing 14(1), 49–62 (2005)

[2] Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Patt. Analysis Machine Intelligence 24(5), 603–619 (2002)

[3] Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: Proc. of IEEE International Conference on Computer Vision and Pattern Recognition, June 2005, vol. II, pp. 1124–1131 (2005)

[4] Deng, Y., Manjunath, B.: Unsupervised segmentation of color-texture regions in images and video. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(8), 800–810 (2001)

[5] Duarte, A., Sánchez, A., Fernández, F., Montemayor, A.: Improving image segmentation quality through effective region merging using a hierarchical social metaheuristic. Pattern Recognition Letters 27(11), 1239–1251 (2006)

[6] Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. IEEE Int. Conf. on Computer Vision, vol. II, pp. 416–423 (2001)

[7] Monteiro, F.C.: Region-based spatial and temporal image segmentation. PhD thesis, Faculdade de Engenharia da Universidade do Porto, Portugal (2007), http://repositorio.up.pt/aberto/handle/10216/11088?mode=full

[8] Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)

[9] Yu, S., Shi, J.: Multiclass spectral clustering. In: Proc. of IEEE International Conference on Computer Vision, Nice, France, October 2003, pp. 313–319 (2003)

# On the Robustness of Fuzzy-Genetic Colour Contrast Fusion with Variable Colour Depth

Heesang Shin, Alwyn Husselmann, and Napoleon H. Reyes

Institute of Information and Mathematical Sciences, Massey University, Auckland
New Zealand
H.Shin@massey.ac.nz, ahusselmann@gmail.com, N.H.Reyes@massey.ac.nz
http://www.massey.ac.nz/~hshin

**Abstract.** Present in this paper is a hybrid Fuzzy-Genetic colour classification system that works under spatially varying illumination intensities, even for moving source illuminants, in real-time. At the heart of the system is an algorithm called Fuzzy Colour Contrast Fusion (FCCF) that compensates for all confounding effects in the environment. We extended FCCF to become self-calibrating using a Heuristic-Assisted Genetic Algorithm (HAGA), and enhanced it further using a technique called Variable Colour Depth (VCD). We devised an improved fitness function for finding the best colour contrast rules and compared it with the rule scoring system used previously by FCCF. Moreover, we tested the integrated algorithms on the FIRA robot soccer platform, but with much more challenging lighting conditions. Our experiments include real-time colour object recognition under extreme illumination conditions, such as, multiple source illuminants, arbitrarily moving source illuminant and colour classification under environments not seen during training. Our results attest to the robustness of the proposed hybrid system, with colour object recognition accuracy ranging from 84% to 100%, measured as robot recognition per frame.

## 1 Introduction

Under spatially varying illumination conditions, coloured objects become difficult to classify algorithmically, as the colours fluctuate throughout the environment. Even worse, when the source illuminants are permitted to move as well as the target object, colour object recognition becomes more complex. Additionally, effects created by spectral reflectance of the target object's colours and the spectral power distribution of the illuminant and characteristics of the camera make colour classification, and hence object recognition difficult.

Presented within this paper is a collection of algorithms that operate in tandem (however not necessarily concurrently) to solve the problem of accurately classifying colours in a real-time environment. Unlike other fuzzy approaches [1,2], our fuzzy system performs colour corrections and works even in the presence of moving source illuminants. The *raison d'être* of the techniques

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 667–674, 2009.

is to foster a more efficient search method for optimal colour classifiers. The approach described in this paper includes further developments upon earlier techniques [3,4], which have already been proven as accurate colour classification models, suitable for vision systems that require real-time image segmentation by colour. Additionally, a system was developed to visualise the effectiveness of the algorithms by utilising physical robots in a controlled environment. Such algorithms are crucial to mobile robot applications [5] like Robocup [6] and FIRA [7].

The algorithms automatically determine the gradient of illumination across the regions, and together, they compensate for varying light intensity at each pixel of the target object. Such an object may consist of any number of colours, but is typically constrained for ease of calibration. In this context, light intensity is a product of the reflectance and illumination intensity at the same point on an object, while ignoring the constant factor created by the arrangement of the optical imaging device [8].

Collectively, the greatest prevalent asset of the techniques is the ability to allow all possible colour combinations to be classified and stored in a variable colour-depth look-up table structure for *real-time* object tracking. The extensions we propose in this paper focus on the self-calibrating Heuristic-Assisted Genetic Algorithm , the new colour contrast rule scoring function and experiments on extreme lighting conditions.

### 1.1 Colour Space and the Pie-Slice Decision Region

The colour descriptors were extracted from a modified rg-chromaticity colour space. The original rg-chromaticity space is already known to remove ambiguities due to illumination or surface pose [9]. However, it is not suitable for pie-slice colour classification [10], and so we modified it. rg-Hue corresponds to the angle, while rg-Saturation corresponds to the radius of a colour pixel.

rg-chromaticities: $r = \frac{R}{R+G+B}$, $g = \frac{G}{R+G+B}$
rg-Saturation $= \sqrt{(r - 0.333)^2 + (g - 0.333)^2}$, rg-Hue $= tan^{-1}(\frac{g-0.333}{r-0.333})$

### 1.2 Fuzzy Colour Contrast Fusion (FCCF)

Colour classification via a pie-slice decision region is hampered by chromaticity distortions due to the confounding effects of spatially varying illumination conditions. These however, are compensated for via a fusion of colour contrast operations employed through fuzzy rules. The inputs are the original colour tristimulus in RGB form, as well as the calculated rg-Hue and rg-Saturation values. The HAGA derives the set of optimal colour contrast rules and the levels of operations (i.e. 1x, 2x, 3x). This also includes the colour contrast constraint angles. Then the appropriate rules can be applied independently on the colour channels. Consequently, FCCF produces the colour-corrected RGB values through the aplication of colour contrast operations defined by Eqns. (1) and (2).

*Example 1.* **Sample colour contrast rule with VCD**
*If(pixel depicts LIGHTBLUE according to colour contrast constraint angles) Then apply colour contrast enhance on the Red channel 2 times, at depth 7, apply colour contrast degrade on the Green channel at depth 6 and apply colour contrast enhance on the Blue channel 3 times at depth 8.*

Contrast Enhance Operator:

$$\alpha = \begin{cases} 2{\mu_\alpha}^2(y) & 0 \leq \mu_\alpha(y) < 0.5 \\ 1 - 2[1 - \mu_\alpha(y)]^2 & 0.5 \leq \mu_\alpha(y) \leq 1 \end{cases} \tag{1}$$

Contrast Degrade Operator:

$$\alpha = \begin{cases} 0.5 + 2[\mu_\alpha(y) - 0.5]^2 & 0 \leq \mu_\alpha(y) < 0.5 \\ 0.5 - 2(1 - [\mu_\alpha(y) + 0.5]^2) & 0.5 \leq \mu_\alpha(y) \leq 1 \end{cases} \tag{2}$$

### 1.3 Variable Colour Depth

In the RGB colour space, three components are usually represented using equal magnitudes (i.e. 0..255), and therefore, represented using equal number of bits. On the other hand, in the Variable Colour Depth representation, [4,11] each colour component could be represented using a varying number of bits. For example, a colour depth of 6-bits for red, 8-bits for green, and 8-bits for blue in the RGB colour space means that the red component has a quarter less resolution than the green and blue components. In this research, we are not reducing the colour depth of the image merely for reduced storage purposes, but we are reducing the colour depth of the image for improved colour classification. The algorithms that we employ aim at increasing colour discriminability of the target objects, especially for cases where there are similar colours present in the scene and they need to be classified accurately. For real-time execution, a special Variable Colour Depth Look-up Table (VCD LUT) is utilised.

### 1.4 General Variable Colour Depth - FCCF System Architecture

All possible colour values in a given Variable Colour Depth is tested in the FCCF component to construct the VCD LUT. Figure 1 shows how the VCD LUT is used in a real-time environment. The acquired colour pixel in the scene is converted into a separate Variable Colour Depth representation for each colour classifier. Next, each colour classifier accesses its own VCD LUT to determine the pixel's colour class.

The VCD LUT differs from the standard indexed LUT in that it allows for varying bit numbers in representing each colour component (e.g. 3-bit for red, 8-bits for green and 8-bits for blue). This requires a separate LUT for each predefined colour class, but even so, the size of each VCD LUT is very small as it only requires to hold a truth value for referencing a colour value. Altogether, a collection of VCD LUTs would still be smaller than one standard indexed LUT.

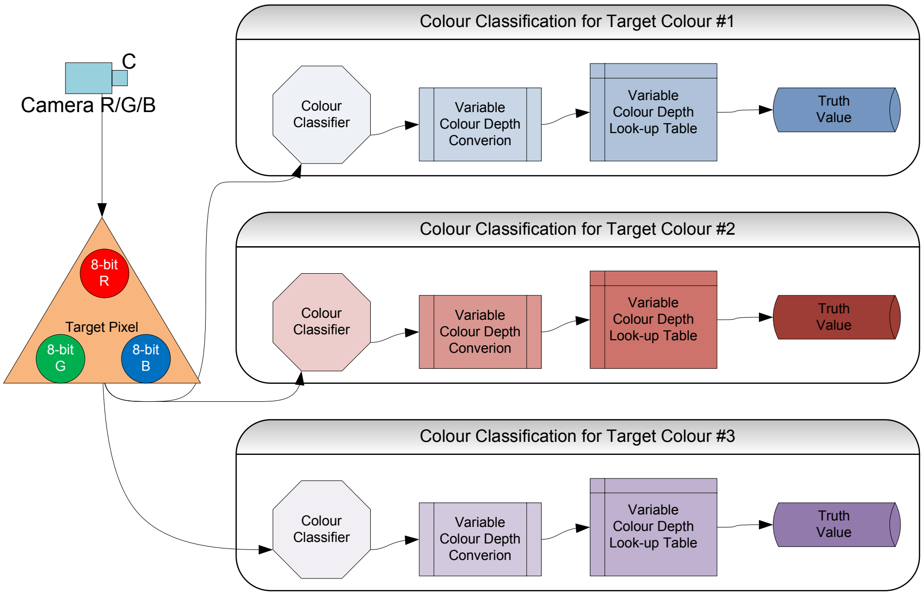

C
Camera R/G/B
8-bit R
Target Pixel
8-bit G
8-bit B
Colour Classification for Target Colour #1
Colour Classification for Target Colour #2
Colour Classification for Target Colour #3
Colour Classifier
Variable Colour Depth Convertion
Variable Colour Depth Look-up Table
Truth Value

**Fig. 1.** Variable Colour Depth Look-Up Table for Real-Time Processing

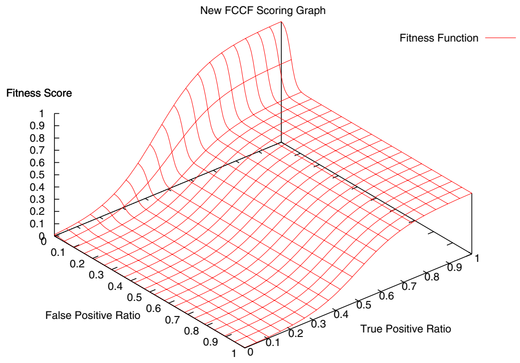
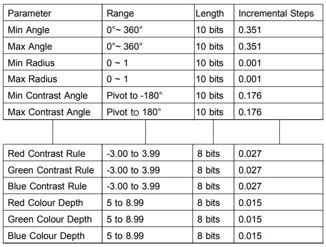


**Fig. 2.** 3D Mapping of Fitness Function



| Parameter | Range | Length | Incremental Steps |
|---|---|---|---|
| Min Angle | 0°~ 360° | 10 bits | 0.351 |
| Max Angle | 0°~ 360° | 10 bits | 0.351 |
| Min Radius | 0 ~ 1 | 10 bits | 0.001 |
| Max Radius | 0 ~ 1 | 10 bits | 0.001 |
| Min Contrast Angle | Pivot to -180° | 10 bits | 0.176 |
| Max Contrast Angle | Pivot to 180° | 10 bits | 0.176 |
| Red Contrast Rule | -3.00 to 3.99 | 8 bits | 0.027 |
| Green Contrast Rule | -3.00 to 3.99 | 8 bits | 0.027 |
| Blue Contrast Rule | -3.00 to 3.99 | 8 bits | 0.027 |
| Red Colour Depth | 5 to 8.99 | 8 bits | 0.015 |
| Green Colour Depth | 5 to 8.99 | 8 bits | 0.015 |
| Blue Colour Depth | 5 to 8.99 | 8 bits | 0.015 |

**Fig. 3.** Chromosome Design

## 2 Fuzzy-Genetic Colour Classifier Extraction

### 2.1 Motivation

The search space to be explored in finding an optimal colour classifier in FCCF is vast due to the real number valued-parameters of classification, such as contrast angles and radii to mention a few. The colour classifier requires a large number of parameters to calibrate. This leads to a lot of difficulty in generating an optimal colour classifier automatically. There are 6 real number-valued parameters (classification angles, contrast angles and classification radii), 3 sets of classification operations and 3 sets of Variable Colour Depth subranges which all affect the result of colour classification independently. The HAGA offers to find a solution from the search space by performing mutation and crossover operations on the chromosomes. The algorithm may end up with a non-optimal solution. However, it is highly likely to return a more accurate set of colour classifier parameters than the manually calibrated ones.

### 2.2 Chromosome Design

The chromosome defines the search space of the Genetic Algorithm. As discussed in the earlier sections, there is a total of 12 different parameters to construct a colour classifier; thus, there are also 12 different parameters to optimise. We designed the chromosome with a total size of 108 bits (Fig. 3). The chromosome design is largely divided into two sections. The front 60 bits correspond to angles and radii, while the last 48 bits correspond to contrast rules and colour depth values. We divided it into two because the latter 48 bits could be disabled when using the guided search strategy.

For the front 60 bits of the chromosome, 4 angles and 2 radii are assigned each with a 10-bit range sub-chromosome. Each sub-chromosome could represent $2^{10}$ values, which ranges from 0 to 1 representing the radius. This is about 0.001 incremental steps for the radius. If the angle parameter is using the full 0 to 360 range, the increments are about 0.35 degrees. However, this can be sliced more narrowly if we limit the search range for the angles. In the experiments, we limited the search range for the contrast angle up to 180 degrees origin from each side of classification angle. This allows incremental steps of about 0.176 degree for contrast angles.

The last 48 bits of the chromosome, divides into a length of 8-bit segments for representing the integral values of the contrast rules and colour depths. Since 8 bits are somewhat larger than the required 7 states of colour contrast rules and 4 levels of colour depths, it allows for larger variances of crossover and mutation operations.

### 2.3 Fitness Function for Colour Classification

A new fitness function is introduced to increase the performance of the Genetic Algorithm. Compared to the previous fitness function [12], the new fitness function adopts a logistic function to evaluate a candidate classifier's performance more accurately. The fitness function awards 1.0 for perfect colour classification, and is totally independent of the structure and/or the number of colour classifier parameters. This is a very desirable feature for a fitness function. The new fitness function as shown in Eqn. (3) is designed to adaptively allow false positive classifications in order to attract more true positive classifications. In order to avoid getting trapped in local maxima, rewards are minimised in cases where false positives and true positives are very low. In contrast, the new fitness function increases fitness values when the true positive ratio gets higher, whereas the old fitness function is weighted more on minimising false positives provided a quarter of each target area is classified correctly.

$$\begin{aligned} x &= \frac{\textit{true positive pixels in target area}}{\textit{total pixels in target area}} \\ y &= \frac{\textit{false positive pixels in outside target area}}{\textit{total pixels outside target area}} \\ fitness &= \frac{1}{2}\left[\frac{1}{e^{-10(y-0.5)}} + \left(\frac{1-\frac{1}{1+e^{-75(x-0.05)}}}{1+e^{-10(y-0.4)}}\right)\right] \end{aligned} \tag{3}$$

# 3 Real-Time Robot Tracking Experiment

## 3.1 Experimental Platform

The new fitness function was assessed on the same soccer board test bed used in [12] for comparison purposes. The calibration setup is non-typical, as it is plagued with spatially varying illumination intensities. During calibration, five target colours, represented by 45 colour patches, were deliberately positioned to be exposed under these varying degrees of illumination intensity. Performance of real-time colour classification is measured by counting the number of frames and the number of frames in which the robot was successfully classified.

Some of the constituents of the testing platform are beyond the scope of this paper. This includes the navigation control systems used to guide the robot to a target destination, and object recognition algorithms. Relevant calibration on these systems were *not* carried out. This resulted in some unintended, sub-optimal paths being generated.

The experiment is arranged as three tests, each with different illumination conditions. During each of the three tests, the colour classification parameters were not amended in any way. The resultant image of classification on static objects is shown in Fig. 4. A blue circle and a white pixel count indicates an area that was classified, and together with the 8-connected components algorithm [13], identified as a cohesive colour segment.

For the experiments we used a single non-incandescent 24W, 1450 lumen, white light (6500K colour temperature). This light presents a very slight shade of blue.

# 4 Results and Analysis

Only ambient flourescent light was present in the first test. The resultant final image produced is shown in Fig. 5. As the colour classifiers were not calibrated for this environment, it was expected that the success rate for this test would not be perfect. Each yellow dot on the picture in Fig. 5 indicates the centre of

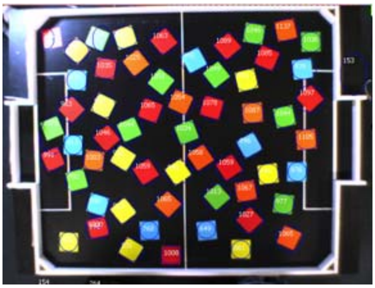


**Fig. 4.** Static object recognition test

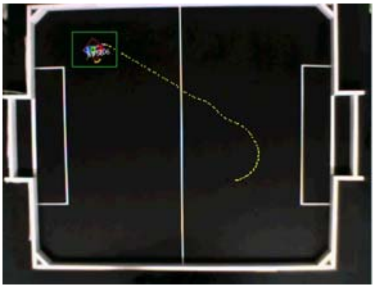


**Fig. 5.** Final resultant image generated after Test 1

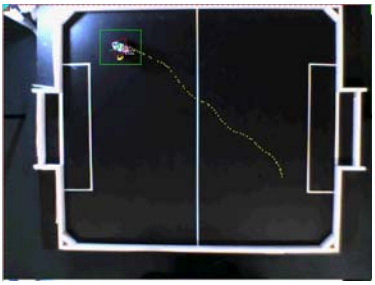


**Fig. 6.** Final resultant image generated after Test 2

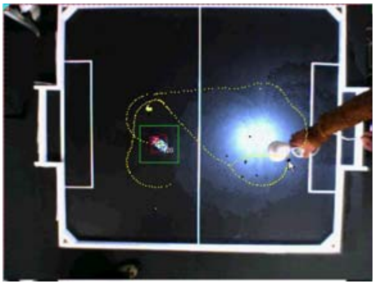


**Fig. 7.** Final resultant image generated after Test 3

the robot at a particular point on the path that it took to the target point. The test resulted in 85% accuracy for 5 tests on average.

The second test was performed with one fixed light pointing directly towards the centre of the board. This test yielded a 100% success rate, for 5 tests on average. The final image generated after this test is shown in Fig. 6.

The third and final test was performed with one moving light at an average of 3.0 ft from the robot. Such a test has not been previously carried out with FCCF. Such an issue is an example of an extreme case of lighting conditions faced by vision systems that were designed with real-time colour segmentation in mind. The robustness of the classifiers generated by the HAGA and the new fitness function could be demonstrated very well using this test. Alternating target points were provided to keep the robot moving while the test was being carried out. The success rate for this test was 97.2%, for an average of 5 tests.

## 5 Conclusions and Future Work

In this paper, we have introduced a collection of algorithms that work in real-time. The algorithms include FCCF, VCD and a HAGA which execute only once to calibrate a special VCD look-up table that is used for real-time classification.

Even with moving source illuminants, inherent varying spatial illumination and presence of similar colour segments (eg. orange and red) a high degree of accuracy is achieved. Movement of coloured target objects further compounds the burden of precision classification due to the gradient of illumination that needs to be compensated for by the system. Colour object classification accuracy using the FIRA robot soccer environment under extreme dynamic, ambient and fixed illuminance is between 84% and 100%.

## References

1. Kashanipour, A., Milani, N.S., Kashanipour, A.R., Eghrary, H.H.: Robust color classification using fuzzy rule-based particle swarm optimization. In: Li, D., Deng, G. (eds.) Cisp 2008: First International Congress on Image and Signal Processing, Proceedings, Sanya, Peoples R China, May 27-30, vol. 2, pp. 110–114. IEEE Computer Soc., Los Alamitos (2008)
2. Hildebrand, L., Fathi, M.: Knowledge-based fuzzy color processing. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 34(4), 499–505 (2004)
3. Reyes, N.H., Dadios, P.E.: Dynamic color object recognition using fuzzy logic. Journal of Advanced Computational Intelligence and Intelligent Informatics 8, 29–38 (2004)
4. Shin, H., Reyes, N.: Variable colour depth look-up table based on fuzzy colour processing. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009, Part II. LNCS, vol. 5864, pp. 1071–1078. Springer, Heidelberg (2009)
5. Bruce, J., Balch, T., Veloso, M.: Fast and inexpensive color image segmentation for interactive robots. In: Proceedings of IROS 2000, pp. 2061–2066 (2000)
6. Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., Matsubara, H.: Robocup: A challenging problem for ai. AI Magazine 18, 73–85 (1997)
7. Kim, J.H., Seow, K.T.: Soccer Robotics. Springer, Heidelberg (2004)
8. Horn, B.K.P.: Determining lightness from an image. Computer Graphics and Image Processing 3, 277–299 (1974)
9. Finlayson, G.D., Schiele, B., Crowley, J.L.: using color for image indexing. In: Proceedings of Challenge of Image Retrieval (1998)
10. Thomas, P., Stonier, R., Wolfs, P.: Robustness of color detection for robot soccer. In: Proceedings of the Seventh International Conference on Control, Automation, Robotics and Vision, pp. 1245–1249 (2002)
11. Shin, H.: Finding near optimum colour classifiers: Genetic algorithm-assisted fuzzy colour contrast fusion using variable colour depth. Master's thesis, Massey University (2009)
12. Playne, D.P., Mehta, V.D., Reyes, N.H., Barczak, A.L.C.: Hybrid fuzzy colour processing and learning. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part II. LNCS, vol. 4985, pp. 386–395. Springer, Heidelberg (2008)
13. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. PWS Publishing (1999)

# Navel Orange Blemish Identification for Quality Grading System

MingHui Liu[1], Gadi Ben-Tal[1], Napoleon H. Reyes[2], and Andre L.C. Barczak[2]

[1] Compac Sorting Equipment LTD, 11 String Street,
PO Box 13 516, Onehunga, Auckland 1643, New Zealand
{steven,gadi}@compacsort.com
[2] Massey University, Albany Campus, Computer Science Department,
Auckland, New Zealand
{n.h.reyes,a.l.barczak}@massey.ac.nz

**Abstract.** A novel automated blemish detection system for ripe and unripe oranges is proposed in this paper. The algorithm is unique in that it does not rely on the global variations between pixels depicting the colours of an orange. By utilizing a priori knowledge of the properties of rounded convex objects, we introduce a set of colour classes that effectively 'peels-off' the orange skin in order of increasing intensity layers. These layers are then examined independently, allowing us to scrutinize the skin more accurately for any blemishes present locally at the layer's intensity variation range. The efficacy of the algorithm is demonstrated using 170 images captured with a commercial fruit sorting machine as the benchmarking test set. Our results show that the system correctly classified 96% of good oranges and 97% of blemished oranges. The proposed system does not require any training.

**Keywords:** Orange Blemish Identification, Fruit Grading, Image Processing.

## 1 Introduction

Orange is an important horticultural produce around the world amounting to millions of tons per annum and is projected to grow by as much as 64 million by 2010 [1]. Their value however is greatly affected by fruit diseases and damages due to transport and handling. Traditional inspection of fruits by human experts is considered to be very time-consuming and subjective [2, 3]. On the other hand, there are not many robust and accurate grading systems evaluating fruit defects comprehensively. Therefore, quick, accurate, consistent and more elaborate automated defect inspection systems are of paramount importance to the growing market.

Several machine vision systems were suggested for sorting oranges [4-10]. However, most of the existing algorithms are not able to explicitly mark the pixels corresponding to the blemishes, but could only provide a final answer (i.e. good or bad orange). Moreover, most of these systems are not easily customizable to meet the evolving requirements set by the market because they rely on extensive training with thousands of examples [11]. There are some commercially available fruit sorting

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 675–682, 2009.

machines that can be modified quickly to meet changing requirements [12] but those generally requires a supervisor monitoring the machine.

Upon examining the aforementioned algorithms, a novel system for grading oranges into different quality bands, according to their surface characteristics, is devised and presented in this paper. Both ripe and unripe oranges comprise the benchmarking dataset. It was observed that unripe oranges are more difficult to analyze for defect detection due to the colour transition areas. In addition, global variation between pixels from the same orange is deemed not to be sufficient to classify defects correctly. Most of the existing algorithms disregard this significant issue. However, the novel algorithm takes full advantage of the global intensity variation for blemish detection purposes. We provide evidence of the merits of using this global intensity variation in our experiments.

## 2 The Algorithms

There is a limit to most existing orange blemish detection algorithms. Any two pixels in an orange image having about the same colour will almost always be classified as belonging to the same category (either a blemish or not). This however presents a big problem, as depicted in Figure 1, it is possible to have several pixels depicting more or less the same colour channel values, but they should belong to different categories. In the figure, pixel A reflects R=134, G=86, B=24 and should be classified as a normal skin. On the other hand, Pixel B is described to have colour channel values very close to Pixel A, but should be classified as belonging to a blemish. Note that this problem stems from the illumination of the fruit and does not change when using a different colour space such as HSI. One potential solution is to use an edge detector to find the discontinuities in colour and detect blemishes based on the difference between neighboring pixels. This approach however tend to detect 'false blemishes' on unripe (green) oranges where a sharp discontinuity may occur between green and orange spots on the fruit. In light of this problem, this research utilizes a priori knowledge of the local intensity variation observed on rounded convex objects to classify the aforementioned pixels correctly.

For any rounded convex object, the intensity gradually increases from the edges to the center in a two-dimensional image as showed in Fig. 1. The proposed algorithm

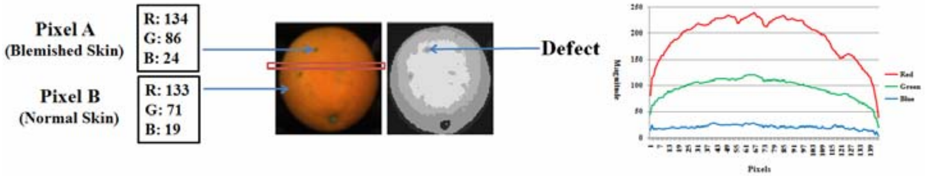


**Fig. 1.** Blemish detection based on a specified local image variation range. Original image of an orange (left) and partitioning (right) defects are visible as 'holes' in the top partition. With pixel values of blemished and good regions and RGB values variations along a line crossing the fruit (far right).

partitions the given image into eight orange colour classes. This in turn would generate different layers/classes using average intensities for a given image (illustrated in Fig. 1). These layers are then refined further to eliminate extraneous layers. Finally, the blemishes are detected by employing a convex hull approach on the topmost layer. Any discontinuities between successive layers/classes will be detected as blemishes.

### 2.1 System Architecture

A block schematic of a typical fruit grading system is shown in Fig. 2. This paper expands the blemish detection part of the system. Stem detection, blemish quantification and grading are left for other research.

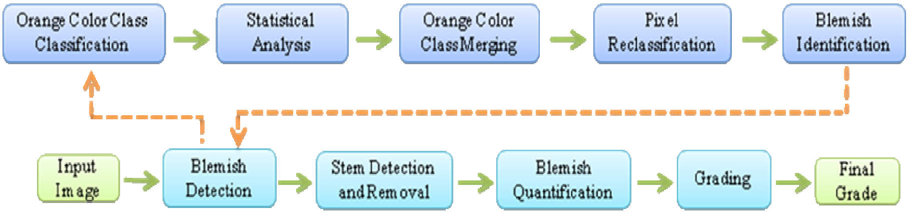


**Fig. 2.** Block schematic of the novel orange grading algorithm

The novel blemish detection algorithm simulates how humans make observations of the local intensity variations phenomenon. Humans do not judge the colours of the orange skin by using absolute pixel values per se (i.e. RGB) but instead consider the neighboring orange surface characteristic. The stem and navel of the orange will be detected as blemishes and a second pass looking at the identified blemishes will be needed to exclude these parts from the total defect score for the fruit.

Quantifying blemishes is a necessary precursor to grading the oranges. Here, the percentage of blemishes over the whole orange is computed as the main grading feature. In addition, the different quality bands can be adjusted easily according to the requirement set by the market.

### 2.2 Blemish Detection

The heart of the algorithm deals with partitioning with a set of thresholds. On a clean orange without blemishes, these thresholds will generate a set of 'concentric rings', with each ring being of a darker colour as we move away from the centre of the orange. Blemishes, however, will be seen as pixels belonging to a dark ring inside a brighter ring as shown in the right side image of Fig. 1. Section 2.2.1 explains an algorithm for generating these thresholds, while Section 2.2.2 explains how the 'rings' are used for blemish detection.

### 2.2.1 Orange Partitioning

*2.2.1.1 Orange Colour Classes.* Otsu's thresholding method [13] is used here independently for each of the colour channels to automatically find a threshold that will isolate the orange fruit from the background. Otsu's thresholding method assumes that the image to be thresholded contains two classes of pixels (i.e. foreground=1 and background=0). Using the three thresholds, each pixel can be classified into one of 8 colour classes using the following formula.

$$ClassNo_{(i,j)} = 4Red_{(i,j)} + 2Green_{(i,j)} + Blue_{(i,j)} + 1$$
$$for\ 1 \leq i \leq Rows, 1 \leq j \leq Columns \quad (1)$$

Each of these orange colour classes produces different results in colour and brightness. For instance, the combination of red and green colour is greater than the combination of blue and green colour. For other fruits, the order of the orange colour classes has to be changed accordingly to match the incremental sequence. Due to the observed nature of the orange skin colours, the different orange colour class described above is ordered incrementally according to their average magnitudes.

After classifying all the pixels comprising the entire orange image according to the 8 orange colour classes, the next objective is to merge the classes according to a similarity measure based on the following: orange class mean per channel, standard deviation per channel, squared mean difference (SMD) between colour classes per channel and quadratic mean of the combined SMD for all RGB channels, quadratic mean of the combined standard deviation for all RGB channels. The merging of the colour classes will eventually lead to the extraction of the topmost layer that will serve as the main focus of inspection.

*2.2.1.2 Class Statistics.* The mean and standard deviation (SD) of each channel inside each class are calculated.

$$\mu = \frac{\sum_{i=1}^{n} P_i}{n} \quad (2)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (P_i - \mu)^2}{n}} \quad (3)$$

*Where P* is the pixel value from one colour channel and *n* is the number of pixels.

Squared mean difference SMD is defined as follows:

$$SMD = (\mu_A - \mu_B)^2 \quad (4)$$

*where* A *and B represents any one of the 8 orange colour classes, and* A ≠ B.

SMD is a measure of the squared difference between two class means. A small-valued SMD indicates that the two classes are similar to each other.

The class statistics are all tripled for all 3 channels. These are all converted into a single number by taking the quadratic mean (QM) of the value among all 3 channels.

$$QM_V = \sqrt{\frac{\Sigma V^2}{3}} \tag{5}$$

*2.2.1.3 Closest Neighbor Class.* Finding the closest neighbor class requires computing for the minimum between the QM-SMD values of one class against the rest of the other classes, and its own QM-SD. If QM-SD < QM-SMD, then the class is its own closest neighbor.

*2.2.1.4 Class Merging.* After determining the closest neighbor for each class, the classes are merged together with their closest neighbor and their corresponding class mean are recalculated. There is a possibility however that some classes will not be merged with other classes, and some classes might also disappear when their class mean is zero. At the end of this stage we will have up to 8 classes. For each class an upper threshold is calculated as the average between this class mean and the next class mean.

$$T_i = \frac{(M_{i+1} + M_i)}{2} \tag{6}$$

And each pixel is then reclassified using the set of new thresholds.

### 2.2.2 Blemish Pixels Identification

*2.2.2.1 Top Layer Slicing.* The top layer slicing phase of the algorithm is important as this defines the region of inspection. After class merging and pixel reclassification, the newly-derived orange colour classes organize itself automatically in an incrementing intensity fashion. An example of which is shown in Fig. 3 - a grey scale image generated after pixel reclassification is depicted in the figure.

Only the topmost layer is analyzed and the reasons are explained as follow:

1. The lighting condition is better on the top layer of the orange.
2. Noise is filtered out, such as background pixels.
3. The orange is rotated on the conveying system, so unprocessed parts can be analyzed in the next image as there are 25 images taken for each orange fruit.
4. Unnecessary computations are reduced for unstable data.

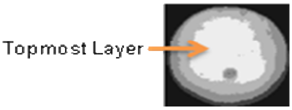


**Fig. 3.** Topmost layer

*2.2.2.2 Blemish Segmentation.* The blemishes are identified by employing a convex hull approach [14] on the topmost layer. The convex hull of the top layer is calculated and then the original layer is subtracted from the convex hull. The resulting pixels are identified as blemishes as seen in Fig. 4. To avoid false positives around the edge of the top layer, a filtering step is added before convex hull calculation.

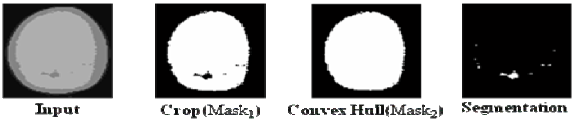


**Fig. 4.** Blemish segmentation on the topmost layer

# 3 Results

Fig. 5 shows some sample blemishes detected using this algorithm for ripe and unripe oranges. A simple grading scheme was implemented grading each image by the percentage of pixels that are classified as blemished. If at least 1% of the pixels were blemished, then the image is considered to reflect a blemished fruit. Using this classification requirement, 100 images of good oranges and 70 blemished ones were tested and the results are in table 2 which indicates high classification accuracy.

**Table 1.** Summary of classification results using the novel algorithm

| Fruit Image Types | No. of Images | No. of Correctly Classified Images | No. of Wrongly Classified Images | % of Correct Classification |
|---|---|---|---|---|
| Good Orange | 100 | 96 | 4 | 96% |
| Blemished Orange | 70 | 68 | 2 | 97% |

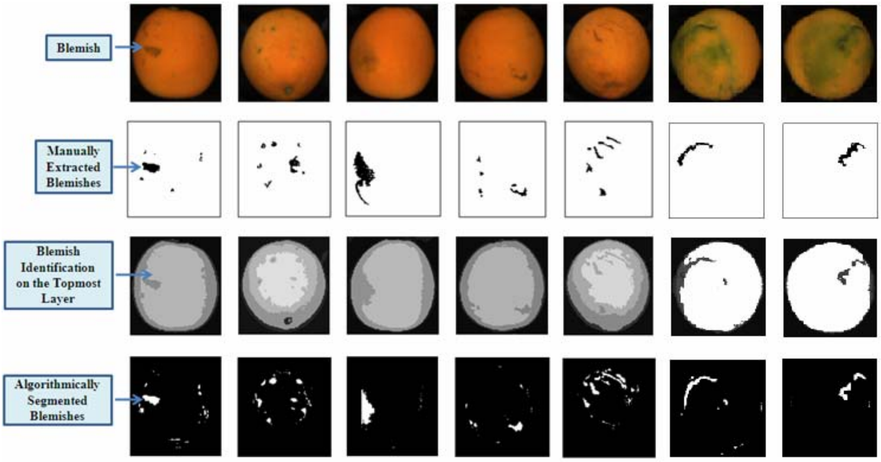

Blemish
Manually Extracted Blemishes
Blemish Identification on the Topmost Layer
Algorithmically Segmented Blemishes

**Fig. 5.** Blemish identification for ripe and unripe oranges. The first row contains five ripe and two unripe blemished oranges. The second row contains manually extracted blemishes by vision experts. The third row contains processed images generated by the novel algorithm. The fourth row contains the segmented blemishes.

## 4 Conclusion and Future Work

An automated intelligent blemish detection system for ripe and unripe oranges is proposed in this research. The main impetus that led to the development of the novel algorithm is the observation that global variations between pixels depicting the same orange fruit are not sufficient for classifying defects. Most existing fruit grading algorithms are not addressing this significant issue and one solution to the problem is presented here. Usual approaches are also plagued with rigorous and lengthy training requirements. On the contrary, the proposed algorithm does not rely on any computationally expensive training. Lastly, some existing algorithms are able to grade fruits into different quality bands (i.e. histogram-based analysis, etc.). However, such algorithms cannot locate explicitly where the blemishes are on the fruit. On the other hand, the algorithm presented here is able to locate the blemish and measure its area with high level of accuracy. This enables a second stage classifier (not discussed in this paper) to process the potential blemishes and classify them in order to eliminate normal feature (stem and calyx ends) and to assign each blemish with a severity index for further grading of the fruit. In the current implementation all defect types contribute roughly equally to the final grading decision. However it would be fairly easy to add a severity index to types of defects. The grade is computed as a measure of the size of these surface defects over the whole orange.

For future works, we envisage that the novel algorithm may be suitable for grading some other fruits that have the rounded convex surface property. However, the orange colour class proposed here will have to be modified slightly to be adapted for grading other fruits. Moreover, an extension of the proposed research would be to improve the grade calculation by incorporating some measure of severity for the blemishes, based on their relative intensities. In effect, it is generally true that the darker the blemishes are, the more severe the damage is.

## References

1. Thomas, H.: Projections of world production and consumption of citrus to 2010 (2009), `http://www.fao.org/DOCREP/003/X6732E/x6732e02.htm#2` (Retrieved June 1, 2009)
2. Brosnan, T., Sun, D.: Improving quality inspection of food products by computer vision – a review. Journal of Food Engineering 61(1), 3–16 (2004)
3. Chen, Y., Chao, K., Kim, M.: Machine Vision technology for agricultural applications. Computers and electronics in Agriculture 36(2-3), 173–191 (2002)
4. Recce, M., Taylor, J., Piebe, A., Tropiano, G.: High speed vision-based quality grading of oranges. In: Proceedings of the 1996 International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing, pp. 136–144 (1996)
5. Unay, D.: Multispectral image processing and pattern recognition techniques for quality inspection of apple fruits. PhD's Thesis, Faculté Polytechnique de Mons, Belgium (2005)
6. Du, C., Sun, D.: Recent developments in the applications of image processing techniques for food quality evaluation. Trends in Food Science Technology 15(5), 230–249 (2004)
7. Vijayarekha, K., Govindaraj, R.: Citrus fruit external defect classification using wavelet packet transform textures and ANN. In: IEEE International Conference on Industrial Technology, pp. 2872–2877 (2006)

8. Guo, F., Cao, Q.: Study on colour image processing based intelligent fruit sorting system. In: Fifth World Congress on Intelligent Control and Automation, vol. 6, pp. 4802–4805 (2004)
9. Bharati, M., Liu, J., John, F.: Image texture analysis: methods and comparisons. Chemometrics and Intelligent Laboratory Systems 72(1), 57–71 (2004)
10. Chang, W., Chen, S., Lin, S., Huang, P., Chen, Y.: Vision based fruit sorting system using measures of fuzziness and degree of matching. IEEE International Conference on Systems, Man, and Cybernetics 3, 2600–2604 (1994)
11. Egmont-Petersen, M., Ridder, D., Handels, H.: Image processing with neural networks. Pattern Recognition 35(10), 2279–2301 (2002)
12. Compac Sorting Equipment LTD, `http://www.compacsort.com`
13. Zhang, J., Hu, J.: Image segmentation based on 2D Otsu method with histogram analysis. In: 2008 International Conference on Computer Science and Software Engineering, vol. 6, pp. 105–108 (2008)
14. Preparata, F.P., Hong, S.J.: Convex hull of finite sets points in two and three dimensions. Communications of the ACM 20(2), 87–93 (Retrieved from Potral) (1977)

# A Cyclostationarity Analysis Applied to Scaled Images

Babak Mahdian and Stanislav Saic

Institute of Information Theory and Automation of the AS CR
Pod Vodárenskou věží 4, 18208 Prague 8, Czech Republic
{mahdian,ssaic}@utia.cas.cz
http://zoi.utia.cas.cz/image-forensics

**Abstract.** The knowledge of image's geometric history plays an important role in image signal compression, image registration, image retrieval and especially in image forensics. In this paper we focus on scaling and show that images that have undergone scaling contain hidden cyclostationary features. This makes possible employing the well–developed theory and efficient methods of cyclostationarity for a blind analyzing of the history of images in respect to scaling transformation. To verify this, we also propose a cyclostationarity detection method applied to our problem and show how the traces of scaling can be detected and the specific parameters of the transformation estimated. The method is based on the fact that a cyclostationary signal has a frequency spectrum correlated with a shifted version of itself. A quantitative measure of the efficiency of the method is proposed as well.

**Keywords:** Interpolation; scaling; cyclostationary; authentication; image forensics.

## 1 Introduction

The knowledge of image's geometric history plays an important role in image signal compression, image registration, medical image analysis, image retrieval, digital publishing, etc. Furthermore, it has a crucial role in image forensics. Here the detection of traces of geometrical transformations signifies photo tampering (for example, see [10,6]). Moreover, when applying a statistical–based method to an image, without knowing the processing history of this image and how the statistics of the image has been changed, we can expect miscalculations and unexpected results. The core of the paper is based on the cyclostationarity theory, which is an attractive and novel one for the computer vision and pattern recognition community.

The term cyclostationarity refers to a special class of signals which exhibit periodicity in their statistics. In this paper, we focus on scaling and show that images that have undergone such a transformation contain hidden cyclostationary features. This will justify employing the well–developed theory of cyclostationarity and its efficient methods for analyzing images' history in respect to geometrical

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 683–690, 2009.

transformations. The cyclostationarity is brought into the signal by the interpolation process (nearest neighbor, linear, cubic, etc.). The interpolation process is present in almost every image resizing operation. Interpolation has a long history and probably started being used as early as 2000BC by ancient Babylonian mathematicians[1]. Despite this long history, the massive usage of interpolation and its importance in digital signal processing, to our knowledge, there exist only a few published works concerned with the specific changes brought into the signal by this process [10,7,5]. The knowledge that interpolated/resampled images are defacto cyclostationary signals makes possible a new point of view to such images and justifies employing existing efficient cyclostationarity detectors to improve the results of mentioned methods.

One of the most important properties of a cyclostationary signal is the existence of specific correlation between its spectral components [3]. Based on this knowledge, we also propose a cyclostationarity detection method capable of determining whether a given digital image is result of scaling. If so, the method also can determine the specific parameters of transformation.

Comparing the method described in this paper with [10] shows, that the latter one is based on a complex and time–consuming EM estimation. Our method uses a simple and fast method for detecting cyclostationary features and achieves very similar results. Furthermore, the output of the method in [10] and the convergency of their EM part directly depend on several initialization parameters. Our method does not need any parameters initialization and work in a complete blind way.

## 2 Cyclostationarity

In the last half a century a lot of work has been done in the field of cylcostationarity [4]. Much of the initial work introducing and examining the use of cyclostationary models in the signal analysis was carried out by W. A. Gardner et al. [2,1,3].

A zero–mean signal $f(x)$ is defined to be second order cyclostationary if its second order statistics are periodic. The autocorrelation function of $f(x)$ can be defined as

$$R_f(x,\delta) = E\{f(x)f^*(x+\delta)\}, \tag{1}$$

because of its periodicity in $x$, we can represent it in the form of a Fourier series expansion:

$$R_f(x,\delta) = \sum_{\alpha} R_f^{\alpha}(\delta)e^{j2\pi\alpha x}, \tag{2}$$

where $\alpha$ is the cyclic frequency. The parameter $R_f^{\alpha}$ is called Cyclic Autocorrelation Function (CAF) and it is a fundamental parameter of cyclostationarity.

[1] For instance, it had an important role in astronomy which in those days was all about time–keeping and making predictions concerning astronomical events [8].

CAF is defined as:

$$R_f^{\alpha}(\delta) = \lim_{X \to \infty} \frac{1}{X} \int_{-X/2}^{X/2} R_f(x, \delta) e^{-j2\pi\alpha x} dx \tag{3}$$

An appropriate way of analyzing cyclostationary properties is by applying the Fourier Transform (FT) to $R_f^{\alpha}$. The result is called Spectral Correlation Function (SCF) and is defined as:

$$S_f^{\alpha}(u) = \int_{-\infty}^{\infty} R_f^{\alpha}(\delta) e^{-j2\pi u\delta} d\delta \tag{4}$$

As we will deal with discrete signals, the discrete version of CAF should also be defined here:

$$R_f^{\alpha}(l) = \lim_{N \to \infty} \frac{1}{N} \sum_{m=0}^{N-1} f[m] f^*[m+l] e^{-j2\pi\alpha m \Delta m}, \tag{5}$$

where $N$ and $\Delta m$ denote the number of samples of the signal and sampling interval, respectively. Equivalently, the discrete SCF can be obtained by:

$$S_f^{\alpha}(u) = \sum_{l=-\infty}^{\infty} R_f^{\alpha}(l) e^{-j2\pi u l \Delta l} \tag{6}$$

CAF and SCF are analogous to the autocorrelation function and power spectral density function for stationary signals. When $\alpha = 0$, the SCF can also be interpreted as power spectral density of the signal. For other values of $\alpha$, SCF is cross–spectral density between the signal and the signal shifted in frequency by $\alpha$. So, if the signal being analyzed exhibits cyclostationarity, SCF will be non–zero for some $\alpha \neq 0$. Otherwise, only for $\alpha = 0$ we will have non–zero values.

## 3 Detecting Cyclostationarity in Scaled Images

We will assume the following simple, linear and stochastic model and assumptions:

$$f(x) = (u * h)(x) + n(x) \tag{7}$$

where $f$, $u$, $h$, $*$, and $n$ are the measured image, original image, system PSF, convolution operator, and random variable representing the influence of noise sources statistically independent from the signal part of the image ($E\{n(x)\} = 0$). If we consider the first part of equation (7) to be deterministic, the covariance of equation (7) can be shown to be $R_f(x_1, x_2) = Cov\{f(x_1), f(x_2)\} = E\{(f(x_1) - \overline{f}(x_1))(f(x_2) - \overline{f}(x_2))\} = Cov\{n(x_1), n(x_2)\} = R_n(x_1, x_2)$, where $R_f$ is the covariance matrix of measured image $f(x)$, and $R_n$ is the covariance of random process $n(x)$.

By $f_k$ we will denote a discrete signal representing the samples of $f(x)$ at the locations $k\Delta$, $f_k = f(k\Delta)$, where $\Delta \in \mathcal{R}^+$, is the sampling step and $k \in \mathcal{N}_0$.

There are two basic steps in scaling transformation. In the first step a spatial transformation of the physical rearrangement of pixels in the image is done. Coordinate transformation is described by a transformation function, $T$, which maps the coordinates of the $x^{'} = T_x(x,y), y^{'} = T_y(x,y)$.

The second step is the interpolation step. Here pixels intensity values of the transformed image are assigned using a constructed low-pass interpolation filter, $w$. As the word interpolation signifies[2], the interpolation process can be described using the following convolution:

$$f^w(x) = \sum_{k=-\infty}^{\infty} f_k w(\frac{x}{\Delta} - k) \tag{8}$$

Many interpolation kernels have been investigated and proposed so far [9]. We will be concerned with following low-order piecewise local polynomials: nearest-neighbor, linear, cubic and truncated sinc. These polynomials are used extensively because of their simplicity and implementation unassuming properties. As will be shown, these interpolators bring noticeable periodic artifacts into the signal.

As was shown in [11,7], when considering the stochastic model and conditions stated in the beginning of this section, the variance of an interpolated signal becomes:

$$var\{f^w(x)\} = var\{f^w(x + \vartheta\Delta)\}, \vartheta \in \mathcal{Z} \tag{9}$$

In other words, $var\{f^w(x)\}$ is periodic over x with period $\Delta$. Thus, interpolated signals contain periodically varying properties.

Now, we can employ the theory of cyclostationarity for finding the traces of scaling. Many efficient methods capable of detecting cyclostationary features [4,12] have been proposed so far.

Theory of cyclostationarity has shown that a cyclostationary signal has a frequency spectrum that is correlated with a shifted version of itself [1]. Based on this, we on focus detecting the traces of cyclostationarity by estimating the spectral correlation function. To estimate the SCF, we can simply use equation (6). But, due to its computational complexity, instead of this, we rather use a more computationally effective SCF estimation method based on Fast Fourier Transform (FFT). FFT algorithm has computational complexity O(nlog$_2$n).

Let's say $f(x,y)$ is the image being analyzed and $F(n,u)$ is a matrix containing FFT of image's rows (i.e., $F(1,u)$ contains the one–dimensional FFT of the first row of $f(x,y)$). The SCF can be estimated in the following way:

$$S_f^{\alpha}(u) = \frac{1}{N} \sum_{n=0}^{N-1} F_{n,u} \cdot F^*_{n,u+\alpha}, \tag{10}$$

[2] The word "interpolation" originates from the Latin word "inter", meaning "between", and verb "polare", meaning "to polish" [8].

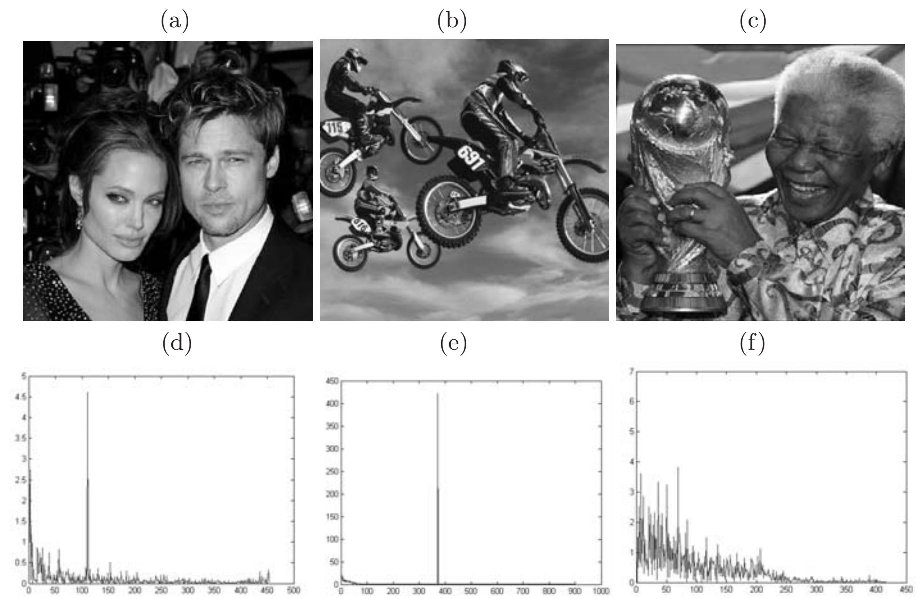


**Fig. 1.** Each column shows an example image and a corresponding correlation map obtained by application of equation (11) to the resized version of this image. The used scaling rates were (from left to right) 1.32. 1.7 and 1.2. In other words, Figure 1 (e) was obtained by applying the method to resized version (with scaling rate 1.7) of Figure 1 (b). In all cases the bicubic interpolation has been used. The distinctive peaks signifying the resizing procedure.

where $*$ denotes complex conjugate and $N$ is the number of image's rows.

Data obtained can be combined together to create the resulting correlation map:

$$\rho_f(\alpha) = \sum_u |S_f^\alpha(u)|^2 \tag{11}$$

To demonstrate the method's output obtained using equation (11) we apply it to several images resized by various scaling factors, see Figure 1. Here, to get clear peaks, columns of $F(n, u)$ were scaled to have values between 0 and 1. As apparent from Figure 1, cyclostationary features resulting from the scaling procedure are exhibited by distinctive peaks.

### 3.1 Adaptation of the Proposed Method

The method presented so far works well (produces clear peaks) when the scaling rate is big enough to introduce strong correlation into the image. When the image is transformed by a lower scaling rate, the cyclostationary features are not strong enough to be detectable using the basic method (see Figure 1 (c)). This drawback can be overcome using a traditional way based on passing the

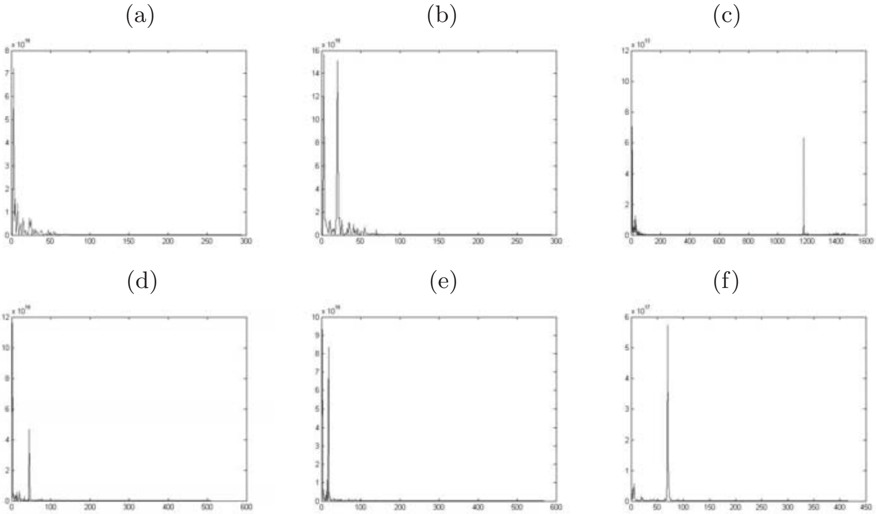


**Fig. 2.** A few examples of the method's output. In (a) the output of the image applied to the non–resized version of Figure 1 (a) is shown. As apparent, there is no clear peak. (b) and (c) show the output of the method for resized Figure 1 (a) with scaling rates 1.07 and 4.11 respectively. (d) and (e) show the output of the method for resized Figure 1 (b) with scaling rates 0.92 and 1.03. In (e) the output of the method applied to the resampled version of Figure 1 (c) with scaling rate 1.2 is shown. In all cases the bicubic interpolation has been used.

analyzed image through a set of band–pass filters. We use a set of derivative filters as band-pass filters. If $f(x,y)$ denotes the image being analyzed, $\mathbf{d_x}$ and $\mathbf{d_y}$ are band–passed images containing the horizontal and vertical derivative approximations:

$$\mathbf{d_x} = [-1 \quad 1] * f(x,y) \qquad \mathbf{d_y} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} * f(x,y) \tag{12}$$

Applying the equations (10) and (11) to $\mathbf{d^n}$,

$$\mathbf{d^n} = \mathbf{d^n_x} + \mathbf{d^n_y}, \tag{13}$$

where $\mathbf{n}$ denotes the order of derivative filter, results in significantly more accurate and robust outcomes. After performing various experiments with resized images of different structures, brightness and noise characteristics we achieved good results using only a lower and a higher derivative filter. The proposed method consists of application of equation (10) to $\mathbf{d^1}$ and $\mathbf{d^5}$ separately:

$$\begin{aligned} \rho_{\mathbf{d^1}}(\alpha) &= \sum_u |S^{\alpha}_{\mathbf{d^1}}(u)|^2 \\ \rho_{\mathbf{d^5}}(\alpha) &= \sum_u |S^{\alpha}_{\mathbf{d^5}}(u)|^2 \end{aligned} \tag{14}$$

If the image being analyzed has been scaled, then at least one of the correlation maps $\rho_{\mathbf{d}^1}(\alpha), \rho_{\mathbf{d}^5}(\alpha)$ will exhibit a detectable peak (see Figure 2). In all examples, the method has been applied to the green color band.

Detected peaks are directly related with the scaling rate. So, using the position of the occurred peak, the particular scaling rate can be estimated. For example, when assuming the upsampling operation and $a$ will denote the position of the peak in interest obtained by applying the method to $f(x, y)$, then the scaling rate can be estimated in the following way: $s = \frac{y}{y-a}$.

## 4 Quantitative Experiments

The method has been applied to 1000 images undergone various scaling transformations. The size of the investigated images was 512 × 512 pixels. All experiments were carried out in Matlab. In all cases never–compressed images have been used. Table 1 shows the detection accuracy of the method applied to bicubic resized images. The detection accuracy expresses the success of the method in expressing the interpolation by a clear and easily detectable peak either in $\rho_{\mathbf{d}^1}(\alpha)$ or $\rho_{\mathbf{d}^5}(\alpha)$. Note that the detection is nearly perfect for scaling rates greater than 0.90. Here, the amount of the cyclostationary features is strong enough to be detectable. When the image is downsampled, the power of cyclostationary features brought into the signal is weakened and a lot of information is lost (due to downscaling). This makes the detection of downsampling difficult. Shown statistics for scaling rate 1 (non–resized) correspond to the false positives rate of the method. Here, if no peaks are found, the image is denoted as non–resized.

**Table 1.** Detection accuracy [%]. Each cell corresponds to the average detection accuracy from 1000 images.

| scaling rate | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 | 0.97 | 1.00 | 1.03 | 1.05 | 1.10 |
|---|---|---|---|---|---|---|---|---|---|---|
| accuracy | 24 | 91 | 93 | 98 | 100 | 100 | 95 | 100 | 100 | 100 |
| scaling rate | 1.20 | 1.40 | 1.50 | 1.60 | 1.70 | 1.80 | 1.90 | 2.00 | 2.10 | 2.20 |
| accuracy rate | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

## 5 Conclusions and Further Research

The aim of this paper was to show that images that have undergone a geometric transforation contain hidden cyclostationary features. This makes possible employing the existing efficient cyclostationary detectors to find the traces of such transformations. In order to verify this, we focused on scaling and carried out several experiments based on a common cyclostationarity detector. Results obtained are promising and show that employing cyclostationarity methods can be effective. Further research might explore application and evaluation of various types of cyclostationary feature detection methods, use of filter banks and extension to other geometric transformations, such as rotation and arbitrary affine transformations.

**Acknowledgments.** This work has been supported by the Czech Science Foundation under the project No. GACR 102/08/0470.

## References

1. Gardner, W.A.: The spectral correlation theory of cyclostationary time-series. Signal Process 11(1), 13–36 (1986)
2. Gardner, W.A.: Spectral correlation of modulated signals: Part i–analog modulation. IEEE Transactions on Communications 35(6), 584–594 (1987)
3. Gardner, W.A.: Exploitation of spectral redundancy in cyclostationary signals. IEEE Signal Processing Magazine 8(2), 14–36 (1991)
4. Gardner, W.A., Napolitano, A., Paura, L.: Cyclostationarity: Half a century of research. Signal Processing 86(4), 639–697 (2006)
5. Kirchner, M.: Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In: Proceedings of the 10th ACM workshop on Multimedia and security, pp. 11–20. ACM Press, New York (2008)
6. Mahdian, B., Saic, S.: Detection of copy–move forgery using a method based on blur moment invariants. Forensic science international 171(2–3), 180–189 (2007)
7. Mahdian, B., Saic, S.: Blind authentication using periodic properties of interpolation. IEEE Transactions on Information Forensics and Security 3(3), 529–538 (2008)
8. Meijering, E.: A chronology of interpolation: From ancient astronomy to modern signal and image processing. Proceedings of the IEEE 90(3), 319–342 (2002)
9. Meijering, E.H.W., Niessen, W.J., Viergever, M.A.: Piecewise polynomial kernels for image interpolation: A generalization of cubic convolution. In: ICIP (3), pp. 647–651 (1999)
10. Popescu, A., Farid, H.: Exposing digital forgeries by detecting traces of re-sampling. IEEE Transactions on Signal Processing 53(2), 758–767 (2005)
11. Rohde, G., Berenstein, C., Healy, D.: Measuring image similarity in the presence of noise. In: Proceedings of the SPIE Medical Imaging: Image Processing, February 2005, vol. 5747, pp. 132–143 (2005)
12. Serpedin, E., Panduru, F., Sari, I., Giannakis, G.B.: Bibliography on cyclostationarity. Signal Process 85(12), 2233–2303 (2005)

# Non-segmented Document Clustering Using Self-Organizing Map and Frequent Max Substring Technique

Todsanai Chumwatana, Kok Wai Wong, and Hong Xie

School of Information Technology, Murdoch University,
South St, Murdoch, Western Australia 6150
{T.chumwatana,K.wong,H.xie}@Murdoch.edu.au

**Abstract.** This paper proposes a non-segmented document clustering method using self-organizing map (SOM) and frequent max substring mining technique to improve the efficiency of information retrieval. The proposed technique appears to be a promising alternative for clustering non-segmented text documents. To illustrate the proposed technique, experiment on clustering the Thai text documents is presented in this paper. The frequent max substring mining technique is first applied to discover the patterns of interest called Frequent Max substrings or FM from the non-segmented Thai text documents. These discovered patterns are then used as indexing terms, together with their number of occurrences, to form a document vector. SOM is then applied to generate the document cluster map by using the document vector. As a result, the generated document cluster map can be used to find the relevant documents according to a user's query more efficiently.

**Keywords:** Frequent max substring, Self-organizing map, Document clustering.

## 1 Introduction

Due to the rapid increase of the number of electronic documents, document clustering, sometimes referred as text clustering, has become an important task in many fields [1]. Document clustering uses unsupervised learning techniques and may provide fast information retrieval or filtering [2]. This is because clustering categorizes documents into groups based on their similarity in term of their member occurrences. In information retrieval, several thousands documents are often retrieved in response to user's queries. This may sometimes makes it very difficult for users to identify their documents of interest. Clustering is one essential method that helps users to find the relevant document from the collection of documents which are similar to each other and could be relevant to the user's queries. For text clustering in information retrieval, a document is normally considered as a bag of words, although a document consists of a sequence of sentences and each sentence is composed of sequence of words. Very often the positions of words are ignored when performing document clustering. Words, also known as indexing terms, and their weights in documents are usually used

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 691–698, 2009.

as an important variable to compute similarity of documents [3]. The documents contain similar indexing terms and their weight will be grouped under the same cluster. This process is straightforward for European languages where words are clearly defined by word delimiter such as space or other special symbols. European texts are explicitly segmented into word tokens that can be used as indexing terms. Many algorithms have been developed to calculate the similarity of documents and built clusters for fast information retrieval. In contrast, document clustering can be a challenging task for many Asian languages such as Chinese, Japanese, Korean and Thai, because these languages are non-segmented languages, i.e. written continuously as a sequence of characters without explicit word boundary delimiters. Due to this characteristic, members in the document cannot be directly used to calculate the similarity. This suggests that some preprocessing technique may need to be applied first to discover important patterns for Asian documents before clustering can be performed.

## 2 Document Clustering

In document clustering, there are two major clustering approaches: hierarchical and partitional algorithms [4], [5], [6]. Hierarchical clustering approach clusters by producing a nested sequence of partitions that can be represented in the form of a tree structure called a dendrogram. The root of the tree contains one cluster covering all data points, and singleton cluster of individual data point are shown on the leaves of the tree. There are two basic methods when performing hierarchical clustering: agglomerative (bottom up) and divisive (top down) clustering [6]. The advantages of hierarchical approach are that it can take any form of similarity function, and also the hierarchy of clusters allows users to discover clusters at any level of detail. However, this technique may suffer from the chain effect, and it needs more space requirements which are at least quadratic when compare to the *k*-means algorithm. A partitional algorithm [7] is another major clustering approach that can be divided into several techniques e.g., *k*-means [8], Fuzzy c-means [9], QT (quality threshold) [10] algorithms. However, *k*-means algorithm is more widely used among all clustering algorithms because of its efficiency and simplicity. The basic idea of *k*-mean algorithm is that it separates a given data into *k* clusters where each cluster has the center point, also called centroid that can be used to represent the cluster. *K*-data points are randomly selected as the centroids by the algorithm. All data points are then assigned to the closest centroid by computing the distance between every data point and each centroid. Therefore, each centroid and its members can form a cluster. The algorithm also re-computes the centroid of each cluster using the data in the current cluster, and this step is repeated until the centroids stabilize. The main advantages of *k*-means algorithm are its efficiency and simplicity, but it also has several disadvantages. The weaknesses of this technique are that it is only applicable to data sets where the notion of the mean is defined, the numbers of cluster can be identified by users, and it is sensitive to data points that are very far away from other points called outliers [1]. Furthermore, Self-organizing map (SOM) [11], [12] can be used as one of the clustering algorithms in the family of an artificial neural network. As SOM has many successful applications, in this paper we will formulate our technique based on SOM.

## 3 Document Clustering Based on SOM and Frequent Max Substring Mining Technique

In this section, we describe our concept that combines Kohonen's SOM and Frequent max substring mining technique to process the non-segmented text documents into clusters. We will first look at the concept of SOM. SOM is one of the main unsupervised learning methods in the family of artificial neural networks (ANN) that was first developed by Teuvo Kohonen in 1984 [13]. The SOM can be visualized as a regular two-dimensional array of cells or nodes (neurons). The SOM algorithm defines a mapping from the input vector onto a two-dimensional array of nodes. When the input vector $x\in R^n$ is given, it is connected to all neurons in the SOM array denoted as vector $m_i\in R^n$ ,which are associated by each neuron and is gradually modified in the learning process. In mapping, the node where vector $m_i$ is most similar to the input vector $x$ will be activated. This node is often called a best-matching node or a winner. The winner and a number of its neighboring nodes in the SOM array are then tuned towards the input vector $x$ according to the learning principle.

Frequent max substring mining technique was first introduced by Todsanai et al. [14] in 2008. It also has been applied in application for indexing for non-segmented languages [15]. In this paper, we use a set of non-segmented documents (Thai documents) as an input to train a map using SOM. We will describe the process of clustering as follows.

Firstly, the frequent max substring mining technique [14] was proposed to generate frequent max substrings or FMs as a set of indexing terms for a document collection. The input documents are then transformed into a $m*n$ matrix of weighted FM occurrence where $m$ is the number of FMs, and $n$ is the number of documents from the document collection. In that matrix, each column represents a document vector, while each row corresponds to a FM as depicted in figure 1.

$$
\begin{array}{cl}
 & \mathrm{FM_i} \\
V = \left(\begin{array}{ccccccccc}
4 & 3 & 7 & 0 & 0 & 0 & \dots & 0 & 0 \\
3 & 3 & 0 & 0 & 2 & 0 & & 2 & 0 \\
6 & 4 & 5 & 0 & 0 & 0 & & 0 & 0 \\
2 & 2 & 2 & 0 & 0 & 0 & & 0 & 0 \\
3 & 3 & 4 & 0 & 0 & 0 & & 0 & 0 \\
0 & 0 & 0 & 8 & 4 & 7 & & 4 & 9 \\
: & & & & & & & & \\
0 & 0 & 0 & 3 & 0 & 4 & \dots & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 2 & & 0 & 0 \\
0 & 0 & 0 & 4 & 3 & 4 & & 3 & 3 \\
0 & 0 & 2 & 3 & 4 & 3 & & 4 & 0
\end{array}\right)
&
\begin{array}{l}
\text{ผลการแข่งขัน (Competition result)} \\
\text{เป็นอันดับที่ (Competition rank)} \\
\text{ตารางการแข่งขัน (Competition timetable)} \\
\text{กรรมการตัดสิน (Umpire)} \\
\text{รอบรองชนะเลิศ(Semi final round)} \\
\text{อัตราการแลกเปลี่ยนเงินตรา(Currency exchange rat} \\
\quad : \\
\text{ธุรกิจการลงทุน(Investment business)} \\
\text{อสังหาริมทรัพย์(Real estate)} \\
\text{สถานะทางการเงิน (Financial status)} \\
\text{งบประมาณประจำปี (Yearly budget)}
\end{array} \\
\quad\quad d_1\ d_2\ d_3\ d_4\ d_5\ d_i\ \dots\ d_{n-1}\ d_n &
\end{array}
$$

**Fig. 1.** The document vector

Let the set of indexing term FMs = ($FM_{1,}$ $FM_{2,}$ …, $FM_m$) that is extracted as the indexing terms from the set of document $d_j$ in the collection where $j = 1, 2, 3, ..., n$, and $w_{ij}$ is the term frequency of $FM_i$ that appear on document $d_j$.

From figure 1, the vector was used to represent the documents above, and the weight of the document vector is more than 1 if $FM_i$ occurs in the document $d_j$ and 0 if $FM_i$ does not appear in the document $d_j$, i.e.,

$$w_{ij} = \begin{cases} > 1 & \text{if } FM_i \text{ occurs in } d_j \\ 0 & \text{otherwise.} \end{cases}$$

After the transformation process, the document vectors are presented to the map to label the neurons by using SOM. These documents can be labeled into neurons according to their similarity of FM occurrence. Two documents containing the same or mostly overlapping FMs will map to the same neuron. In contrast, the documents may map to distant neurons if they contain different or non-overlapping FMs. Furthermore, the documents with similar FMs may map to neighbouring neurons. This means that the neurons can form the document clusters by examining mapped neurons in the document cluster map. We depict the organization of the document map that clusters similar documents into the same neuron as shown in the boxes. FMs in the boxes represent the content of documents in the collection.

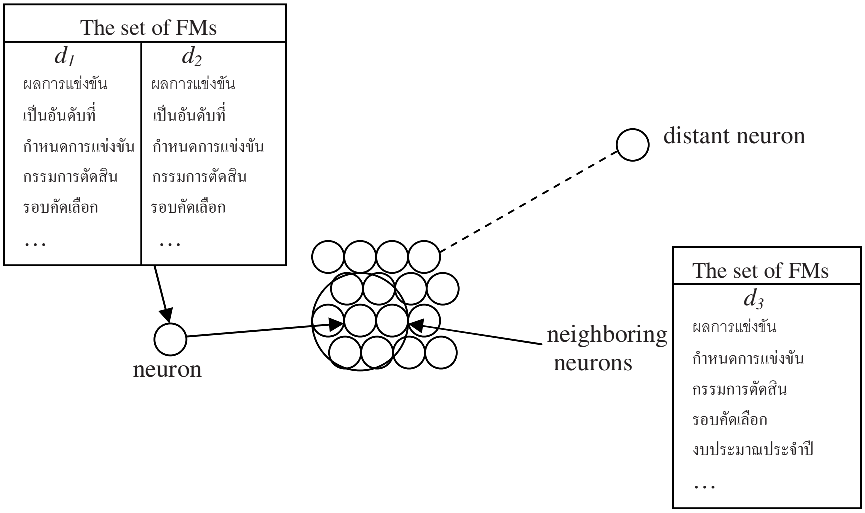


**Fig. 2.** The document cluster map

From the trained map, the document clusters are formed by labeling each neuron that contains certain documents of similar type. The documents in the same neuron may not contain exactly the same set of FMs, but they usually contain mostly overlapping FMs. As a result, the document cluster map can be used as a prediction model to generate the different groups of similar documents, and each group will be then specified the document type by comparing FMs of each group with keywords of each area. In figure 3, we depict clustering the documents into different groups, by mapping an input data with neurons in the document cluster map to find the document groups of several types.

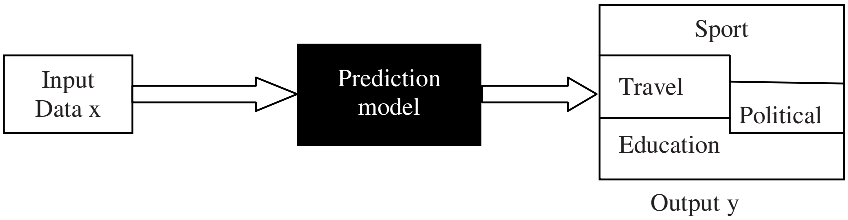


**Fig. 3.** Neuron network architecture

From figure 3, the following will describe the process of matching an input data $x$ with the neurons in the document cluster map by using SOM.

Let us consider the input vector $x = [x_1,\ x_2,\ \ldots,\ x_n]^t \in R^n$ as the input data sets where $t$ is the FMs of the input documents. These input data sets have to be matched with all neurons in the map that is denoted as two-dimensional network of cells or the model vector $m_i = [m_{i1},\ m_{i2},\ \ldots,\ m_{in}]^t \in R^n$ depicted in figure 4. Each neuron $i$ in the network contains the model vector $m_i$, which has the same number of indexing terms as the input vector $x$.

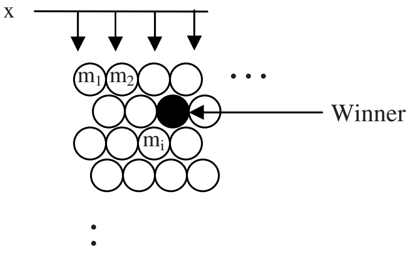


**Fig. 4.** Self-organizing map

From figure 4, the input vector $x$ is compared with all neurons in the model vector $m_i$ to find the best matching node called the winner. The winner unit is the neuron on the map where the set of FMs of the input vector $x$ is the same or most similar to the set of FMs of the model vector $m_i$ by using some matching criterion e.g. the Euclidean distances between $x$ and $m_i$. As a result, this method can be used to cluster documents into different groups, and also it is suggested to reduce the time to search the relevant document in term of information retrieval.

## 4 Experimental Result

In this section, we show an experiment to cluster non-segmented documents (Thai documents) based on using the proposed SOM and frequent max substring technique. 50 Thai documents were used as an input dataset to train a map. All documents can be found on the Thai news websites, which consist of 15 sport, 15 travel, 15 political, and 5 education documents. FM was first generated by frequent max substring technique from the document dataset and 35 FMs, the long and frequently occurring terms in sport, travel, political and education documents, were used as the set of indexing terms for this document collection. 50 input documents are then transformed to a document vector of weighted FM occurrence. Hence, these 35 indexing terms and 50 input documents form a 50 * 35 matrix, where a document vector was represented by each row and each column corresponds to an indexing term. We use this 50 * 35 matrix to train a map using SOM, and the number of neurons was set as 9 in SOM program as shown in figure5. From experimental studies, the group of political, sport, and education documents provided good results. Similar documents of each type were mapped onto the neuron. It can be observed that some errors occured within the group of travel documents. The travel documents were mapped onto several neurons due to overlapping terms that appeared across different type of documents.

The figure 5 showed the map containing 9 neurons and 50 Thai documents. Each neuron contains the group of similar documents.

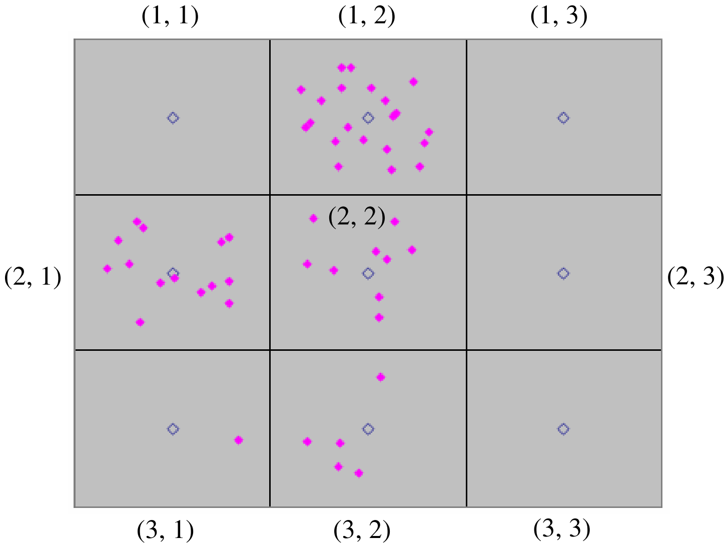


**Fig. 5.** SOM contains 9 neurons and the group of similar documents from 50 Thai document collection

From figure 5, the experimental result showed that SOM can cluster 50 documents into 5 neurons on the map, and the similar documents were grouped into the same neuron as shown in table 1.

**Table 1.** Clustering results of using SOM and Frequent Max Substring technique

| Neuron ID | Row | Column | Document ID |
|---|---|---|---|
| Neuron 5 | 1 | 2 | Political1, Education1, Education2, Education3, Education4, Sport1, Sport2, Sport3, Sport4, Sport5, Sport6, Sport7, Sport8, Sport9, Sport10, Sport11, Sport12, Sport13, Sport14, Sport15, Travel10 |
| Neuron 2 | 2 | 1 | Political2, Political3, Political4, Political5, Political6, Political7, Political8, Political9, Political10, Political11, Political12, Political13, Political14, Political15, |
| Neuron 4 | 2 | 2 | Travel2, Travel4, Travel5, Travel6, Travel7, Travel8, Travel9, Travel13, Travel15, |
| Neuron 1 | 3 | 1 | Travel12 |
| Neuron 3 | 3 | 2 | Education5, Travel1, Travel3, Travel11, Travel14 |

As observed from the results, this technique can be used to cluster non-segmented documents into several groups according to their similarity. The accuracy of this technique is up to 80%. However, from this experiment, we have found that the groups of education and sport documents are mapped onto the same neuron (Neuron5) because they both contain mostly overlapping FMs such as ผลการแข่งขัน (competition result), การจัดอันดับ (position ranking), ได้รับรางวัล (getting award), etc. Furthermore, the contents of documents and generated indexing terms are also the main factors that impact the accurate value. The content of one document may have overlapping terms from two different types of documents. For instance, Education5, Travel1, Travel3, Travel11 and Travel14 documents are mapped onto the neuron 3 because they are presenting information on ecotourism, containing overlapping terms from education and travel documents.

## 5 Conclusions

This paper describes a non-segmented document clustering method using self- organizing map (SOM) and frequent max substring technique to improve the efficiency of information retrieval. We first use the frequent max substring technique to discover FM from Thai text documents, and these FM are then used as indexing terms with their number of occurrences to form a document vector. SOM is then applied to generate the document cluster map by using the document vector. The experiment on clustering the 50 Thai text documents is presented in this paper. From the experimental results, this technique can be used to cluster 50 Thai documents into different clusters with accuracy up to 80%. As a result, the generated document cluster map can be used to find the relevant documents according to a user's query more efficiency. In future work, optimization for the number of clusters and number of neurons in SOM will be studied.

## References

[1] Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer, Heidelberg (2007)

[2] Cutting, R.D., Karger, D.R., Pedersen, J.O., Tukey, J.W.: Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In: SIGIR 1992, pp. 318–329 (1992)

[3] Matveeva, I.: Document representation and multilevel measures of document similarity. In: Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume: doctoral consortium, New York, June 04-09, 2006, pp. 235–238 (2006)

[4] Dubes, R.C., Jain, A.K.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1988)

[5] Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley and Sons, Chichester (1990)

[6] Steinbach, M., Karypis, G., Kumar, V.: A Comparison of Document Clustering Techniques. In: KDD Workshop on Text Mining (2000)

[7] Zhao, Y., Karypis, G.: Comparison of Agglomerative and Partitional Document Clustering Algorithms. In: The SIAM workshop on Clustering High-dimensional Data and Its Applications, Washington, DC (April 2002)

[8] Huang, Z.: Extensions to the K-means Algorithm for Clustering Large Datasets with Categorical Values. In: Żytkow, J.M. (ed.) PKDD 1998. LNCS, vol. 1510, pp. 283–304. Springer, Heidelberg (1998)

[9] Dembele, D., Kastner, P.: Fuzzy C-means method for clustering microarray data. Bioinformatics 19(8), 973–980

[10] Heyer, L.J., Kruglyak, S., Yooseph, S.: Exploring Expression Data: Identification and Analysis of Coexpressed Genes. Genome Research 9, 1106–1115

[11] Fung, C.C., Wong, K.W., Eren, H., Charlebois, R., Crocker, H.: Modular Artificial Neural Network for Prediction of Petrophysical Properties from Well Log Data. IEEE Transactions on Instrumentation & Measurement 46(6), 1259–1263 (1997)

[12] Myers, D., Wong, K.W., Fung, C.C.: Self-organising Maps Use for Intelligent Data Analysis. Australian Journal of Intelligent Information Processing Systems 6(2), 89–96 (2000)

[13] Kohonen, T.: Self-organization and associative memory. Series in Information Science, vol. 8. Springer, Heidelberg (1984)

[14] Chumwatana, T., Wong, K.W., Xie, H.: Frequent Max Substring Mining for Indexing. International Journal of Computer Science and System Analysis, IJCSSA (2008)

[15] Chumwatana, T., Wong, K.W., Xie, H.: Thai Text Mining to Support Web Search for E-commerce. In: The 7th International Conference on e-Business 2008 (INCEB 2008), Bangkok, Thailand (2008)

# A Visual Method for High-Dimensional Data Cluster Exploration

Ke-Bing Zhang[1], Mao Lin Huang[2], Mehmet A. Orgun[1], and Quang Vinh Nguyen[3]

[1] Department of Computing, Macquarie University, Sydney, NSW 2109, Australia
{kebing,mehmet}@science.mq.edu.au
[2] Faculty of Engineering and Information Technology, University of Technology, Sydney NSW 2007, Australia
maolin@it.uts.edu.au
[3] School of Computing and Mathematics, University of Western Sydney, NSW 1797, Australia
vinh@scm.uws.edu.au

**Abstract.** Visualization is helpful for clustering high dimensional data. The goals of visualization in data mining are exploration, confirmation and presentation of the clustering results. However, the most of visual techniques developed for cluster analysis are primarily focused on cluster presentation rather than cluster exploration. Several techniques have been proposed to explore cluster information by visualization, but most of them depend heavily on the individual user's experience. Inevitably, this incurs subjectivity and randomness in the clustering process. In this paper, we employ the statistical features of datasets as predictions to estimate the number of clusters by a visual technique called HOV$^3$. This approach mitigates the problem of the randomness and subjectivity of the user during the process of cluster exploration by other visual techniques. As a result, our approach provides an effective visual method for cluster exploration.

**Keywords:** Cluster Exploration, Visualization, Statistics.

## 1 Introduction

Cluster analysis is an important technique of knowledge acquisition in data mining. To address the requirements of different applications, a large number of clustering algorithms have been developed [9, 3]. However, those algorithms are not very effective in coping with arbitrarily shaped clusters. In addition, cluster analysis is a highly iterative process. However most of existing clustering methods are too automated to exploit the domain experts' knowledge in the intermediate process of clustering. As a consequence, they are not always effective to cluster datasets with a large number of variables and/or huge-sized datasets in real world applications. In a high dimensional space, traditional clustering algorithms tend to break down in terms of efficiency as well as accuracy because data does not cluster well anymore [1].

In order to solve those problems, Shneiderman [19] proposed to present data as a visual plot, so that the user could see the interesting features easily. He pointed out that, visualization can be very powerful and effective in revealing trends, highlighting

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 699–709, 2009.

outliers, showing clusters, and exposing gaps in high-dimensional data analysis. Therefore, the use of visualization to explore and understand high-dimensional datasets is becoming an efficient way to combine human intelligence with the immense brute force computation power available nowadays [16].

Clustering is an exploratory activity [9]. It is an iterative process under the guidance of user domain knowledge. In most cases of the preprocessing stage of clustering, it is hard for the user to estimate the proper cluster number [3]. Visualization is very helpful for the user to do that. However, cluster exploration by visualization mostly depends on the individual user's experience. Thus, subjectivity, randomness and impreciseness may be introduced into the cluster exploration process. As a result, cluster analysis based on imprecise results may be inefficient and ineffective. On the other hand, cluster exploration based on the user's random interaction is arbitrary and it may not be easy to interpret from where the grouped results come.

In this paper, based on the projection of a technique called Hypothesis Oriented Verification and Validation by Visualization (HOV$^3$) [22], we introduce the statistical features of datasets as the predictions of HOV$^3$ to guide the user on cluster exploration, because the statistical summaries objectively reflect the features of datasets. As a result, it provides the user an effective method on determining cluster numbers in the preprocessing stage of cluster analysis.

The rest of this paper is organized as follows. Section 2 briefly introduces cluster visualization techniques and gives a short introduction to the HOV$^3$ technique. Section 3 presents the algorithm of statistics-guided visual approach for cluster detection by HOV$^3$. Section 4 demonstrates the effectiveness of our approach by an experimental analysis on several datasets[1]. Finally, Section 5 summarizes the contributions of this paper.

## 2 Background

### 2.1 Visual Cluster Analysis

Visual cluster analysis is a combination of visualization and cluster analysis. It is believed that the combined strength of visualization and data mining would enrich both approaches and enable more successful solutions [20]. However, the data to be processed by clustering is usually high dimensional. It is not easy to visualize multidimensional data on 2D or 3D space and still give a "genuine" visual interpretation. This is because mapping higher dimensional data onto lower dimensional space inevitably introduces ambiguities, overlapping and even bias. Thus, choosing a technique to fit visualizing clusters of high dimensional data is the first and most crucial task of visual cluster analysis.

In practice, instead of providing a quantitative guidance on cluster exploration, the most of the cluster visualization techniques are typically used as an observational mechanism to assist the user in having intuitive comparisons and understanding of clustering results better. Several approaches have been proposed to help the user on cluster exploration.

[1] The datasets used in this paper are available from http://archive.ics.uci.edu/ml/

For example, Multidimensional scaling (MDS) maps multidimensional data as points into 2D Euclidean space, where the distances between data points reflect the similarity/dissimilarity of them [14]. However, the relative high computational cost of MDS, with polynomial time complexity $O(N^2)$, limits its usability on very large datasets. PCA is a commonly used multivariate analysis technique [10], mainly used for reducing the dimensionality of high dimensional data by extracting the representative variables. However, PCA is sensitive to deal with the non-linear data structure. It is not suitable for the exploration of unknown data. A Grand Tour based visual technique is proposed to visualize cluster structures [5], but this technique visualizes 3 clusters only. To deal with more than 3 clusters with a more sophisticated Grand Tour technique, more assistance is required.

OPTICS uses a density-based technique to detect cluster structures and visualizes them in “Gaussian bumps” [2]. It is an intuitive method to assist the user to observe cluster structures, but its non-linear time complexity makes it neither suitable to deal with very large data sets, nor suitable to provide the contrast between clustering results.

Huang *et. al* [7, 8] proposed several approaches to assist users in identifying and verifying the validity of clusters in visual form. Their techniques work well in cluster identification, but are unable to evaluate the cluster quality very well. On the other hand, these techniques are not well suited to the interactive investigation of data distributions of high-dimensional data sets.

CVAP [21] is a recently proposed prototype with several integrated clustering algorithms and cluster validation methods. It is a convenient toolkit to assist the user on the selection of clustering scheme for the application of small-sized datasets. However, CVAP is only for displaying the clustering and cluster validation results, rather than the purpose for directly evolving the user into the cluster exploration process.

## 2.2 Star Coordinates

The projection of Star Coordinates [11] has only linear time complexity, which is significant for interactive cluster visualization of very large datasets. VISTA [4] and HOV$^3$ [22] extend Star Coordinates by additional features to mitigate the problem of overlapping and ambiguities caused by projecting high dimensional data onto 2D space. The visual approach reported in this paper has been developed based on the projection of HOV$^3$. For the sake of completeness, we briefly introduce the Star Coordinates technique here.

Star Coordinates plots a 2D plane into *n* equal sectors with *n* coordinate axes, where each axis represents a dimension and all axes share the initials at the centre of a circle surface on the 2D space [11]. Star Coordinates first normalizes data in each dimension into a unit interval [0, 1]. Then the values of all axes are mapped to an orthogonal X-Y coordinate which shares the centre point with Star Coordinates on the 2D space. Thus, an *n*-dimensional data item is represented as a point in the X-Y 2D plane by Star Coordinates. Based on this projection, several interaction mechanisms, such as axis scaling, axis rotation, data point filtering are provided in Star Coordinates to change the data distribution of a dataset in order to detect cluster characteristics and render clustering results.

However, it is not easy to give an explanation of the grouping results produced by the user's random interactions in Star Coordinates and VISTA, also the grouping results are usually not repeatable. On the other hand, in the Star Coordinates space, the user's interactions cannot change the data distribution too much when the dimensionality of the dataset is very high (a hundred or more dimensions, which is very common in data mining). This is because the alteration of the data distribution by applying interactions to an axis is much less than that of lower dimensional data in the Star Coordinates space. As a result, in very high dimensional space, it is not effective anymore to separate clusters or explore grouping clues by the interactions of Star Coordinates and VISTA.

As discussed above, the issues of arbitrary exploration and/or complicated visual representation of cluster structures make those techniques inefficient and time consuming on cluster exploration of large and high dimensional data. As Seo and Shneiderman [18] mentioned that "A large number of clustering algorithms have been developed, but only a small number of cluster visualization tools are available to facilitate researchers' understanding of the clustering results". Thus developing an effective visualization technique to assist the user during cluster exploration and detection is the main aim of this research.

## 2.3 HOV$^3$

To remedy the randomness and arbitrariness of visualization on cluster analysis, Zhang *et al.* mathematically generalized the Star Coordinates model by the Euler formula and proposed their visual approach HOV$^3$ to detect clusters [22]. According to the Eular formula: $e^{ix} = \cos x + i \sin x$, where $z = x + i.y$, and $i$ is the imaginary unit. Let $z_0 = e^{2\pi i/n}$; such that $z_0^{1}$, $z_0^{2}$, $z_0^{3}, \ldots,$ $z_0^{n-1}$, $z_0^{n}$ (with $z_0^{n} = 1$) divide the unit circle on the complex 2D plane into $n$ equal sectors. Then Star Coordinates can be simply written as:

$$P_j(z_0) = \sum_{k=1}^{n} [(d_{jk} - \min_k d_k)/(\max_k d_k - \min_K d_k) \cdot z_0^k] \tag{1}$$

where min $d_k$ and max $d_k$ represent the minimal and maximal values of the $k$th coordinate respectively. Equation (1) can be viewed as a mapping from $R^n \rightarrow C^2$.

Conversely, instead of using a random exploration of cluster information by axis scaling or axis rotation in Star Coordinates/VISTA, HOV$^3$ quantifies the user's apriori knowledge/estimation of a studied dataset as a measure vector to precisely guide the user on the exploration of group information. A measure vector $M$ in HOV$^3$ represents the corresponding axes' weight values. Then given a non-zero measure vector $M$ in $R^n$, and a family of vectors $P_j$, the projection of $P_j$ against $M$, according to formula (1), the HOV$^3$ model is presented as:

$$P_j(z_0) = \sum_{k=1}^{n} [(d_{jk} - \min_k d_k)/(\max_k d_k - \min_k d_k) \cdot z_0^k \cdot m_k] \tag{2}$$

where $m_k$ is the $k$th variable of measure $M$ .

It can be observed that, equation (2) is a standard form of linear transformation of $n$ variables, where $m_k$ is the coefficient of the $k$th variable of $P_j$.

## 3 Cluster Exploration by HOV$^3$

We propose a statistics-guided cluster exploration approach by HOV$^3$ based on the following idea. In analytic geometry, the difference of two vectors $A$ and $B$ can be expressed by their inner product $A.B$, with its geometrical meaning that the data distribution is plotted by vector $A$ against vector $B$ (and vice versa). The inner product between a dataset and a measure vector in HOV$^3$ can be geometrically viewed as a data distribution plotted by a set of vectors against the measure vector in the HOV$^3$ space, as shown in equation (2).

Predictive knowledge discovery is an important knowledge acquisition method, which utilizes the existing knowledge to deduce, infer, reason and establish predictions, and verify the validity of the predictions. As mentioned above, the user can quantify his/her priori knowledge of a studied dataset as the guidance on the exploration of group information. Thus the statistical summaries of a dataset can be directly employed as the statistical predictions (measure vectors) of the dataset in HOV$^3$, since the statistical summaries reflect the nature comparisons of data objectively [19]. Also, it is easy to interpret the grouping results of a dataset plotted by statistical predictions in HOV$^3$. The detailed description of our approach is presented below.

### 3.1 The Algorithm

We formalized our idea of using statistical predictions to explore clusters by HOV$^3$ into the algorithm in table 1. The detailed explanation of our algorithm is given next.

**Table 1.** The Algorithm of Statistics-guided Cluster Exploration by HOV$^3$

```
Algorithm: Statistics-guided Cluster Exploration by HOV^3
Input: D: a dataset;  M: statistical measures of D;
Output: G: data distribution of D or subsets of D;
cluster exploration ← true;
p ← D;
$m_i$ ← a statistical measure of p;  ($m_i \in M$)
m ← $m_i$;
while (cluster exploration)
    G ← Hc(D, $m_i$);
    if (G well grouped?)
        if (stop exploration?)
            cluster exploration ← false;
            break;
        endif
    endif
    if (new statistical measure of p?)
        $m_i$ ← a statistical measure of p;
        m ← $m_i$;
    endif
    $m_i$ ← $m \cdot m_i$ ;
endwhile
```

### 3.2 Supported Features

There are two significant features of the use of statistical predictions to explore clusters by HOV$^3$: *Enhanced separation of data groups* and *quantitatively guided exploration*. The projection of HOV$^3$ is simply written as G ← *Hc(D, m)* [23], where *D* is the processing dataset, *m* is a measure vector, and G is the distribution of D projected by HOV$^3$.

- **Enhanced Group Separation**

It is proved that if there are several data point groups that can be roughly separated by applying a measure vector *m* in HOV$^3$ to a dataset, then multiple applications of the projection in HOV$^3$ with the same measure vector to the dataset would lead to the groups being more condensed, i.e., have a good separation of the groups [24].

This feature is achieved by step 6 and step17 in the **while** loop (steps 5-18) of the algorithm, as shown in Table 1. The enhanced group feature is significant for cluster exploration by HOV$^3$ with statistical predictions, since clearly separated groups cannot be usually observed by applying a measure vector to a dataset in HOV$^3$ once.

- **Quantitatively Guided Exploration**

The HOV$^3$ technique provides a quantitative mechanism to visually detect cluster clue by measure vectors. In fact, the statistical summaries of a dataset are quantitative depictions of the dataset. They objectively reflect the natural comparisons of the dataset. Thus introducing them as the predications in HOV$^3$ avoids the randomness and subjectivity which may be introduced by the user during the cluster exploration process by visualization.

To highlight these two features and demonstrate the effectiveness of our approach, we provide several experiments in the next section.

## 4 The Experiments

### 4.1 Parkinson's Disease Dataset

Parkinson's disease dataset has 23 attributes and 195 instances. The original data distribution of Parkinson's disease dataset is shown in Fig. 1, where we cannot recognize any groups in the dataset. Then we choose the standard deviation of the dataset *pstd*=[0.24096, 0.18676, 0.25056, 0.15401, 0.13764, 0.14296, 0.14786, 0.14293, 0.17215, 0.16013, 0.19555, 0.16314, 0.12977, 0.19553, 0.12865, 0.17987, 0.43188, 0.24253, 0.22046, 0.13688, 0.18776, 0.17029, 0.18665] as a statistical prediction to explore the clusters of the dataset. Its projected data distribution is illustrated in Fig. 2, where data points are roughly separated, but we still cannot distinguish groups clearly (3 or 4 groups?).

According to the enhanced separation feature of HOV$^3$ [24], we adopt two times inner product of *pstd* as a statistical prediction and try again. The newly projected result is shown in Fig.3, where the data points are separated into two mains groups, based on the user's observation. We have also used three times mean value of Parkinson's dataset as the statistical prediction to plot the dataset. Its data distribution is shown in Fig.4. It can be observed that, clearly, there are two groups in both Fig.3 and Fig.4.

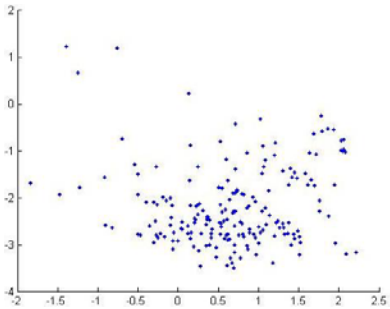


**Fig. 1.** Projecting data distribution by HOV$^3$ of Parkinson's disease dataset without any measurement

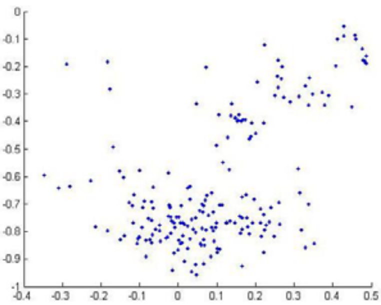


**Fig. 2.** The data distribution projected by HOV$^3$ of Parkinson's disease dataset with its standard deviation, *pstd* as a statistical prediction

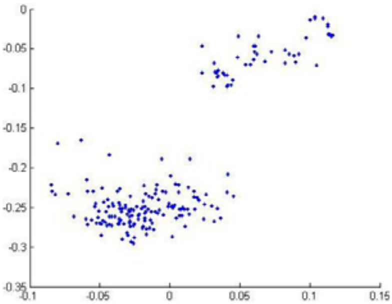


**Fig. 3.** The data distribution projected by HOV$^3$ of Parkinson's disease dataset in Fig.1 with two times of *pstd* as the prediction

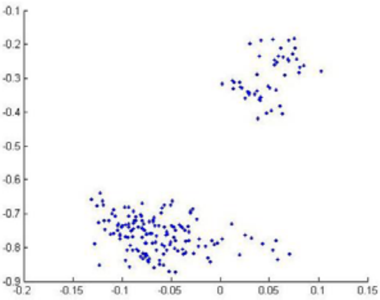


**Fig. 4.** The data distribution projected by HOV$^3$ of Parkinson's disease dataset with three times of mean values of the dataset as a prediction

Based on the above experiments, there are two well-separated clusters in Parkinson's disease dataset. The cluster exploration process can be done iteratively until the user is satisfied by the grouping result by HOV$^3$. He/she can terminate the cluster exploration process by his/her decision (steps 7-12) in table 1.

To verify the validation of the above experiments produced by HOV$^3$, we employed the CVAP system [21] to check the quality of clustering results of Parkinson's disease dataset by K-means [15] and PAM [12] clustering algorithms with a cluster number of 2 to 10. Then we checked the quality of those clustering results by the cluster validation methods of Silhouette index [17] and Dunn index [6]. The higher Silhouette and Dunn indices indicate the better quality of clustering results. The quality tests of those clustering results are illustrated in Fig.5 and Fig.6. It is clear that number 2 is the optimal cluster number of Parkinson's disease dataset for K-means and PAM clustering. This example shows that statistics-guided cluster exploration by HOV$^3$ provides an effective visual method to assist the user on the acquisition of the cluster number in the preprocessing stage of clustering.

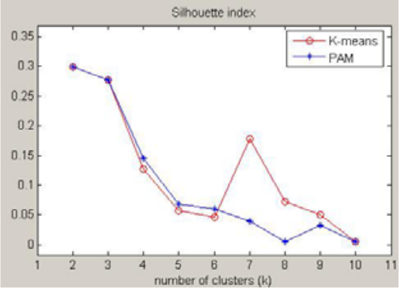


**Fig. 5.** The quality indicated by Silhouette index of clustering results of Parkinson's disease dataset produced by K-means and PAM clustering algorithms with the cluster number ranging from 2 to 10

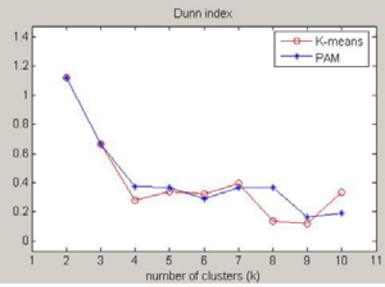


**Fig. 6.** The quality indicated by Dunn index of clustering results of Parkinson's disease dataset produced by K-means and PAM clustering algorithms with the cluster number ranging from 2 to 10

## 4.2 Wine Dataset

We have also applied our approach to the *wine* dataset, which has 13 attributes and 178 instances. Fig.7 and Fig.8 present the original data distribution of the wine dataset and the data distribution projected by HOV$^3$ with three times standard deviation of the dataset respectively. Clearly, there are three well-separated groups in Fig.8. Then we cluster these three groups ($C_H$).

SOM (*Self-organizing Map*) is a neural network based clustering algorithm [13], which has been widely applied in machine learning and data mining. We applied the SOM to the *wine* dataset with cluster number 2-10, and employed the Silhouette index validation algorithm to verify the clustering results in CVAP. Fig.9 illustrates the curve of validation results produced by Silhouette index in CVAP, where we can observe that number 3 is the optimal cluster number of the wine dataset.

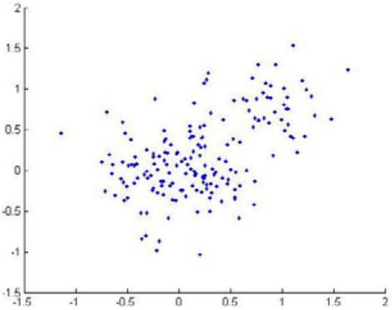


**Fig. 7.** The original data distribution of the *wine* dataset by HOV$^3$

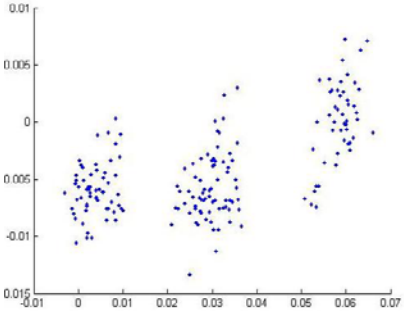


**Fig. 8.** The data distribution projected by HOV$^3$ of the *wine* dataset with its three times of stand deviation values

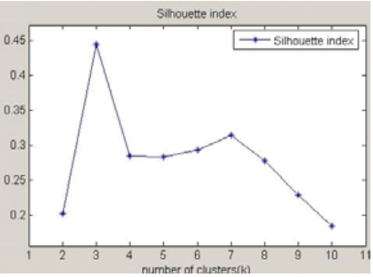


**Fig. 9.** The quality indicated by Silhouette index of clustering results of the wine dataset produced by SOM clustering algorithms with a cluster number of 2 to 10

**Table 2.** The statistical contrast between the clusters (k=3) produced by HOV$^3$ with three times standard deviation of the wine dataset and the clusters produced by SOM clustering algorithms

| $C_H$ | % | *Radius* | *Variance* | *Weighted Variance* | $C_S$ | % | Radius | Variance | *Weighted Variance* |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 26.966 | 102.286 | 0.125 | 3.37075 | 1 | 33.708 | 107.980 | 0.124 | 4.179792 |
| 2 | 39.888 | 97.221 | 0.182 | 7.259616 | 2 | 38.764 | 97.449 | 0.185 | 7.17134 |
| 3 | 33.146 | 108.289 | 0.124 | 4.110104 | 3 | 27.528 | 102.008 | 0.126 | 3.468528 |
| | | | | **14.74047** | | | | | **14.81966** |

The contrast of the clusters ($C_H$) projected by HOV$^3$ and the clustering result ($C_S$) produced by the SOM clustering algorithm is summarized in Table 2. The weighted variance of the two clustering results is listed in the last row of the table. We can see that the quality of $C_H$ is even slightly better than the quality of $C_S$ based on the variance contrast. We believe that a domain expert could give a better and intuitive explanation about this clustering result. This experiment also supports the effectiveness of our approach.

As the examples have demonstrated, visual projection based on the statistical prediction by HOV$^3$ is a more purposeful and effective method for cluster exploration, and also it is easier to obtain a geometrical interpretation of the clustering results.

## 5 Conclusions

We have proposed a statistics-guided visual approach to assist the user during cluster exploration, and demonstrated its effectiveness by experiments on several datasets. This approach adopts the statistical summaries of a high dimensional dataset as predictions to project the data so that the user can have an intuitive observation of clusters during cluster exploration. The use of statistical features of data mitigates the weaknesses of randomness and arbitrary exploration of the existing visual methods employed in data mining. As a consequence, with the features of enhanced group separation and quantitatively guided exploration of our approach, the user can effectively identify the cluster number in the preprocessing stage of clustering.

## References

1. Abul, A.L., Alhajj, R., Polat, F., Barker, K.: Cluster Validity Analysis Using Subsampling. In: Proc. of IEEE International Conference on Systems, Man, and Cybernetics, vol. (2), pp. 1435–1440 (2003)
2. Ankerst, M., Breunig, M., Kriegel, H.-P., Sander, J.: OPTICS: Ordering Points To Identify the Clustering Structure. In: Proceedings of ACM SIGMOD 1999, pp. 49–60 (1999)
3. Berkhin, P.: A Survey of Clustering Data Mining Techniques. In: Kogan, J., Nicholas, C., Teboulle, M. (eds.) Grouping Multidimensional Data, pp. 25–72. Springer, Heidelberg (2006)
4. Chen, K., Liu, L.: VISTA: Validating and Refining Clusters via Visualization. Journal of Information Visualization l3(4), 257–270 (2004)
5. Dhillon, I.S., Modha, D.S., Spangler, W.S.: Visualizing class structure of multidimensional data. In: The 30th Symposium on the Interface: Computing Science and Statistics, vol. (30), pp. 488–493 (1998)
6. Dunn, J.C.: Well Separated Clusters and Optimal Fuzzy Partitions. Journal of Cybern. (4), 95–104 (1974)
7. Huang, Z., Cheung, D.W., Ng, M.K.: An Empirical Study on the Visual Cluster Validation Method with Fastmap. In: The proceedings of DASFAA 2001, pp. 84–91. Springer, Heidelberg (2001)
8. Huang, Z., Lin, T.: A visual method of cluster validation with Fastmap. In: The proceedings of PAKDD 2000, pp. 153–164. Springer, Heidelberg (2000)
9. Jain, A., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323 (1999)
10. Jolliffe, I.T.: Principal Component Analysis. Springer, Heidelberg (2002)
11. Kandogan, E.: Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In: The proceedings of ACM SIGKDD 2001, pp. 107–116 (2001)
12. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data, An Introduction to Cluster Analysis. John Wiley and Sons, Brussels (1990)
13. Kohonen, T.: Self-Organizing Maps, vol. 30. Springer, Heidelberg (1995)
14. Kruskal, J.B., Wish, M.: Multidimensional Scaling, SAGE university paper series on quantitive applications in the social sciences, pp. 07–011. Sage Publications, CA (1978)
15. MacQueen, J.B.: Some Methods for classification and Analysis of Multivariate Observations. In: The proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press, Berkeley (1967)
16. Pampalk, E., Goebl, W., Widmer, G.: Visualizing Changes in the Structure of Data for Exploratory Feature Selection. In: The proceedings of SIGKDD 2003, Washington, DC, USA (2003)
17. Rousseeuw, P.: Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Journal of Computing 20, 53–65 (1987)
18. Seo, J., Shneiderman, B.: From Integrated Publication and Information Systems to Virtual Information and Knowledge Environments. In: Hemmje, M., Niederée, C., Risse, T. (eds.) From Integrated Publication and Information Systems to Information and Knowledge Environments. LNCS, vol. 3379, pp. 232–240. Springer, Heidelberg (2005)
19. Shneiderman, B.: Inventing Discovery Tools: Combining Information Visualization with Data Mining. In: Jantke, K.P., Shinohara, A. (eds.) DS 2001. LNCS (LNAI), vol. 2226, pp. 17–28. Springer, Heidelberg (2001)
20. Westphal, C., Blaxton, T.: Data Mining Solutions: Methods and Tools for Solving Real-World Problems. John Wiley and Sons, Chichester (1999)

21. Wang, K., Wang, B., Peng, L.: CVAP: Validation for Cluster Analyses. Data Science Journal 8(20), 88–93 (2009)
22. Zhang, K.-B., Orgun, M.A., Zhang, K.: HOV$^3$: An Approach to Visual Cluster Analysis. In: Li, X., Zaïane, O.R., Li, Z.-h. (eds.) ADMA 2006. LNCS (LNAI), vol. 4093, pp. 316–327. Springer, Heidelberg (2006)
23. Zhang, K.-B., Orgun, M.A., Zhang, K.: A Visual Approach for External Cluster Validation. In: Proc. of IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007), Honolulu, Hawaii, USA, April 1-5, pp. 576–582. IEEE Press, Los Alamitos (2007)
24. Zhang, K.-B., Orgun, M.A., Zhang, K.: A Prediction-based Visual Approach for Cluster Exploration and Cluster Validation by HOV$^3$. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 336–349. Springer, Heidelberg (2007)

# An Algorithm Based on the Construction of Braun's Cathode Ray Tube as a Novel Technique for Data Classification

Mariusz Swiecicki

Cracow University of Technology, Institute of Computer Modelling, Cracow, Poland
mswiecic@gmail.com

**Abstract.** In this article we have presented a model used for a classification of multidimensional data in a broader sense, called Braun's cathode machine. The internal structure of the machine presented on this paper has been based on the architecture of a cathode-ray tube – Braun's tube. For a machine model described this way a machine training algorithm has been proposed as well as response computing algorithms. In the final chapter we have presented the results of the machine tests for the notions connected with the classification and self-organization of multidimensional data.

**Keywords:** Braun's machine, Classifier algorithm, Self-organization of data algorithm, Data mining, Computational intelligence.

## 1 Introduction

In recent years, there has been an increase of interest in biology as a source of inspiration to tackle many computational problems in the scope of data processing. This interest has been and is motivated by the willingness to select mechanisms that are used by natural systems and the attempt to adjust them to efficient problem solving in the aforementioned areas. The attempts to use elementary physical phenomena in data processing can be an alternative way of searching for natural computational models in relation to biology.

This article presents a data processing mechanism that uses the phenomena occurring during movements of particles in the electric field. The aforementioned occurrences have already been widely used in construction of Braun's cathode ray tube. In the first part of the article physical rights will be presented which have been used in data processing by a processing element called Braun's cathode machine or Braun's machine. In the second part of the article we will discuss the elements of the aforementioned processing element and its functioning mechanism will be presented. In the second part of this article we will also present Braun's cathode machine training algorithm and an algorithm which computes a response. The third part of this article presents the results of the computations carried out with the use of this model.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 710–719, 2009.

## 2 Physical Phenomena Used by Braun's Machine

In this part of the article we will discuss physical phenomena which will constitute a functioning base for Braun's machine. The reaction of electric fields to charged particles is used in two aspects. First of all, it is used to accelerate or curb particle movements, and secondly, to change the direction of moving particles.

### 2.1 Speeding Up the Particles

Let us consider its motion in homogenous electrostatic field with *E* strength. If the direction of *v0* velocity is in accordance with the direction of electric field lines, this field operates on a charged particle with force *F=qE* which is constant so the movement of the particle is a uniformly variable motion. The charge will then accelerate or curb depending on the senses of vectors *v0* and *E*. May a field in *A* point have a potential $V_A$ and in *B* point - a potential $V_B$, so electrostatic potential energy of charge *q* in point *A* equals $qV_A$, and in *B* point, it equals $qV_B$. Appropriate kinetic energies are: in point *A* it is $(mv_0^2)/2$ , in point B $(mv^2)/2$ where the *v* variable is a searched value. Taking advantage of the energy conservation law we obtain [4]:

$$v = \sqrt{v_0^2 + \frac{2q}{m}U} = \sqrt{v_0^2 + \frac{2qEd}{m}} \tag{1}$$

The velocity of the particle is proportional to the element of accelerating voltage. As it results from the equation (1), the velocity value of the particle can be increased by increasing the field strength *E* or by increasing the distance *d* where the acceleration occurs.

### 2.2 Change of the Particle Trajectory

Let's assume that electric field is generated by two plates and the voltage which is led to them is *U*; then *E=U/d*. May the particle of *m* mass possess a *q* charge and initial velocity of *v0* which is perpendicular to force lines. According to the principles of electrostatics, the particle is influenced then by force perpendicular to *v0* of *F=qE* value which will cause the curving of this trajectory according to the equation [4]:

$$y = \frac{qE}{2mv_0^2}x^2 \tag{2}$$

$$v_y = \sqrt{\frac{2qEd}{m}y_0} = \frac{qUl}{mv_0 d} \tag{3}$$

At the escape of the particle from the *x=l* field, and velocity in this place is expressed. Where $v_y$ can be determined based on the equation (1).

## 3 The Architecture of Braun's Machine

In this paragraph we will present a model of a processing element whose functioning uses a physical phenomenon described above. The functioning of Braun's machine is mainly modelled on the principle of functioning of Braun's cathode ray tube, so logical and physical construction of this processing clement is similar to the construction of the tube.

## 3.1 Elements of Braun's Machine

In figure Fig 1. a model of Braun's machine has been presented. Similarly to the tube, the model is equipped with an accelerating grid and a set of deflection plates and a visualizing space. In the described model, a particle gun is responsible for generation of particles. The gun is an equivalent for a cathode in the tube.

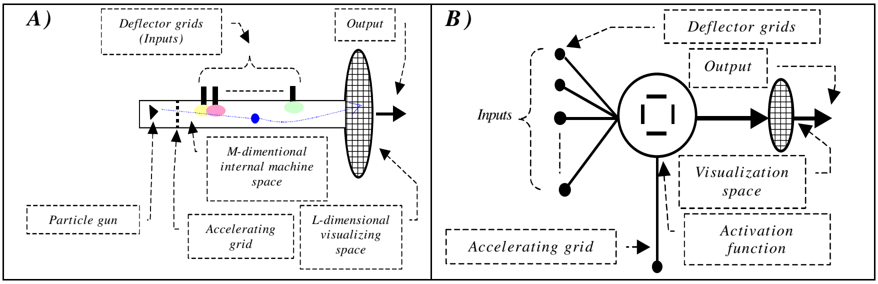


**Fig. 1.** A model of *M*-dimensional Braun's machine with *L*-dimensional visualizing space a) physical construction of a machine b) logical construction of a machine

### Accelerating Plates

Plates are very important elements in the process of data processing. The task of accelerating plates is to accelerate particles which have been released by the gun. The value of the initial velocity of a particle is determined by the value of a control signal led to this plate. This signal corresponds to the accelerating voltage in Braun's cathode ray tube.

### Deflection Plates

A set of deflection plates is the main processing element of the model. The number of deflection plates corresponds to the number of inputs the machine has been equipped with. The size of the input signal led to the input of the machine corresponds to the value of deflecting voltage in a physical equivalent of the machine. The task of deflection plates is to change of the trajectory of a particle. The scope of changes in the trajectory of particles depends on the value of deflecting voltage led to deflection plates (3) which in the model we are describing corresponds to the values of input data.

### Visualizing Surface

The last element of the presented Braun's machine model is the object which is responsible for visualization of the final location of a particle. This object is called a machine's visualizing surface. The functions of this element are the same as the functions of a screen in Braun's cathode ray tube. It can be said that the screen of Braun's tube provides projecting operations of the location of a particle in the three-dimensional space on the plane.

**Data Processing Mechanism**

In the model being described in this paper data processing consists in subsequent modifications of the trajectory of a particle as a result of electric fields which influence the particle. The operation of deflection plates is the source of this influence. The value of electric field intensity is proportional to the value of input data. The higher values of data led to deflection plates we observe, the higher deflection it is. The result of data processing by a machine will be the value of final location of a particle in the space where it moves. Considering the equations (3) it can be stated that the higher velocity of the particle, the smaller influence of electric field of deflection plates we can observe. Therefore, the influence of the values of input data on the processing operation and modification of the trajectory of a practice will be smaller. Increasing the velocity of a particle, we have an influence on generalisation of input data.

### 3.2 A Model of N-Dimensional and M-Input Braun's Machine

After presenting a general idea of functioning of Braun's machine we will draw our attention to creation of a generalized model. Generalization of this model refers to a number of important aspects. The first aspect refers to the relation between the number of machine inputs and the number of deflection plates. It is a very important notion in case if N-dimensional data are to be processed by a machine. The second aspect is connected with the size of the internal machine space in which particles are moving. The size of this space decides on the classification qualities of a machine.

**Construction of a Function of Machine Activation**

The description of machine operation presented in the previous paragraph implies that the process of data processing consists in repeated performing of three types of operations on a particle, namely, an operation of acceleration of deflection and visualisation. Each of the operation mentioned here has an effect on the final location of a particle. In such a case a function of machine activation will be a creation of accelerating function, deflecting function and visualising function. Assuming that the accelerated particles move in *M*-dimensional space, then location vector $X=(x_1,x_2,..x_M)$ and velocity $V=(v_1,v_2,..v_M)$ of a given particle.

*Accelerating Function*

In order to make the processing take place in a machine, a particle should be accelerated to a particular velocity. As it results from the aforementioned description, the value of this velocity decides on the course of the data processing operation performed by machine. The particle acceleration operation is described by means of accelerating function. Definitions of the accelerating function $f_a$ were depicted using the two equations (4) and (5). The adopted notation of $V^i$ indicates that the argument of a given function the i-th component of velocity vector *V*.

$$fa_j : (V_t^i, u_a) \rightarrow V_{t+1}^i \tag{4}$$

$$fa(V_t^i, u_a) := \sqrt{(V_t^i)^2 + K_a u_a} \tag{5}$$

The function (4,5) provides changes of i-th coordinate of velocity vector *V*. As it results from the presented equation (5), velocity that will be gained by a particle because of its acceleration depends on two factors. First of all, it depends on velocity possessed by a particle before entering the area where the grid influences it. Secondly, it depends on the value of accelerating voltage. In the model we are describing, a numerical value is a counterpart of the value of accelerating voltage. In the equation (5) in comparison to the equation (2) there are no values of electric charge of a particle and mass. The reason for that is the fact that these quantities in the presented model have been adopted as constant values and are represented by means of a constant $K_d$.

*Deflecting Function*

Another operation that is performed on a particle during the process of data processing is an operation which modifies the trajectory of a particle motion depending on the value of input data. In a machine, deflation plates carry out this operation. A number of modifications of the trajectory of a particle motion corresponds to the number of deflation plates the machine is equipped with, but the number of plates depends on the dimension of input data and corresponds to the number of machine inputs. The equations (6, 7) define a function which performs the operation of deflection. A deflecting function *fd* possesses two arguments.

$$fd_j : (V_t, u_d) \rightarrow V_{t+1}^i \tag{6}$$

$$fd_j(V_t, u_{d_j}) := \frac{K_d u_{d_j}}{\sqrt{v_1^2 + v_2^2 + \ldots v_M^2}} \tag{7}$$

The first argument is a velocity vector of a particle *V* which it has upon the moment of input in the area of influence of a given deflection plate. The second argument is the value of deflecting voltage $u_d$ which corresponds to i-th element of the input vector. The operation of deflection described by *fd* function modifies the value of velocity vector *V*. During this operation, only this component of *V* vector is modified which belongs to the same plane as given deflection plates. Velocity for the selected coordinate of *V* vector is calculated as a quotient of deflecting voltage and total velocity of a particle. Similarly to the situation connected with a function which accelerates parameters which are constant, i.e. the size of electric charge of a particle, a mass of a particle and the width of deflection plates in formula (7) were not directly taken into account. The value of these parameters is represented by a $K_d$ constant.

*Visualization Function*

The last type of operation that is operated by a machine on a particle is the operation of particle visualization. Similarly to two previously discussed operations, this operation will also be described by means of a function. The operation of particle visualization is realized by means of *L*-dimensional visualizing surface. This surface is the counterpart of a screen covered with luminophore in Braun's cathode ray tube and, similarly to the screen, it is responsible for recording final deflections of a particle against the initial trajectory.

$$fv:(V,(k_1 \times k_2 \times \ldots k_M)) \rightarrow (\overbrace{x_{k_1} \times x_{k_2} \times \ldots x_{k_M}}^{L \subseteq \{1,..M\}}) \tag{8}$$

The formula (8) describes the operation realized by visualizing surface of a machine. Visualization function *fv,* playing the role of a screen, maps a set of values of M-dimensional velocity vector *V* of a particle on its final location *X* in a selected *L*-dimensional subspace which is the counterpart of *L*-dimensional screen.

*Activation Function*

The activity of a machine consists in repeated performance of elementary operations on a particle in motion. Activation function describes the relation between input data and output data. For a Braun's machine we are describing in this paper, activation function is a combination of functions which correspond to elementary operations.

### Algorithm for Computing activation Function

This algorithm computes the location of a particle in a visualizing surface. The location results from the influence of "electric fields" generated by input data of a machine on the particle.

```
Set the input parameters of the algorithm
–   M – dimension of internal machine space; L – dimension of visualizing surface
–   $V=(v_1,v_2,\ldots v_M)$-particle velocity vector;
–   $X_{End}=(x_1,x_2,\ldots x_L)$- location of a particle in a visualizing surface
–   QueueOfPlates – set of deflection and accelerating plates of a machine
–   Input – a set of input data ; $i \leftarrow 0$;
Foreach plate ∈ QueueOfPlates do
    If plate – is an accelerating grid then
        $u_a \leftarrow U_a[plate]$; // collect the value of accelerating voltage for plate
        axis ← SpaceOfMachine[net];// collect the axis along which the plate operates
        $V \leftarrow fa(V, axis, u_a)$; // Perform the operation of particle acceleration
    ElseIf plate – is a deflection plate then
        $i \leftarrow i+1$; // subsequent number of machine input
        $u_d \leftarrow Input[i]$;// take the value of data for a given i-th input.
        axis ← SpaceOfMachine[plate];// take the axis along which the plate operates
        $V \leftarrow fd(V, axis, u_d)$; // Perform the deflection operation
    End if; End foreach
$X_{end} \leftarrow fv(V, (i_1, i_2, \ldots i_L))$;// Perform the visualization operation where $i_1, i_2, \ldots i_L \in \{1,..M\}$
```

**Fig. 2.** Algorithm for computing Braun's machine activation function

In Fig. 2 we presented algorithm for computing the value of activation function. The operation of the algorithm consists in performing elementary operations on a particle that is moving in the machine space. In order to do this, we check the set of plates a machine is equipped with. This set is organized in a form of a queue because it is important to remember that subsequent plates are collected in a specific order that corresponds to a physical location of accelerating and deflection plates inside the machine. After collecting a given plate from the queue of plates, the type of plate is checked. Depending on the type of the selected plate, specific operations for this type are performed. If the plate is an accelerating plate, we collect a value from a given machine input. This value is a counterpart of a deflecting voltage and is an argument of the deflecting function. In the next steps other attributes of a given plate are collected and the deflecting function is computed. When we perform all operations that were determined by the content of the *QueueOfPlates,* we perform visualization operation. This type of operation consists in mapping a velocity vector of a particle on its final location on the machine's visualizing surface.

## Braun's Machine Training Algorithm

Machine training is a one-phase process. This process consists in computation of machine activation function for each vector of input data and in recording the location of a particle in the visualizing surface corresponding to this vector. In Fig. 3 we have presented Braun's machine training algorithm. This algorithm by the analogy to the phenomenon of luminophore burning in the cathode ray tube has been called a "burned surface" algorithm. The training set consists of vectors of input data together with a class number assigned to each vector. The results of the computation of the activation function are the coordinates of a particle on the visualizing surface. A class number is assigned to each location of a particle on a surface and its environment where a given input vector belongs.

```
Set the parameters of training process
–   $TrainSet=\cup\{Input\ x\ \ Class\}$- train set
–   BurnedSurface- visualizing surface with the recorded location of particles
For i=1 to length(TrainSet) do
    $(y_k....y_l)\leftarrow fBM$ (QueueOfPlates, M, L, $TrainSet_i\rightarrow Input$);// Compute activation function
    $BurnedSurface\leftarrow BurnedSurface\cup\{(y_k....y_l)\rightarrow Class\}$;// Allocate class number to the location
End for
```

**Fig. 3.** Braun's cathode machine training algorithm

The algorithm enables generalization of input data in the process of training by means of decrease of influence of deflection plates of a machine as a result of increase of "accelerating voltage".

## Response Computing Algorithm

The algorithm for computing responses was presenteed in Fig. 4. The following elements belong to its parameters: a set of machine's plates that describes its physical structure and burned surface with class numbers belonging to given points on this surface. This algorithm computes activation function for a given data vector. This function returns the coordinates of a particle and its environment on the basis of the physical structure of a machine and the value of input vector.

```
$(y_k....y_l)\leftarrow fBM$(QueueOfPlates, M, L, Input);// Compute activation function
//$(y_k....y_l)$ includes coordinates for an input vector on burned surface
If BurnedSurface$(y_k....y_l)\neq NULL$ then
    $Ans\leftarrow BurnedSurface(y_k....y)$;// return the class numbers belonging to the point and environment $_l$
Else // failed attempt to classify input vector
    $Ans\leftarrow NULL$; // point $(y_k....y_l)$ does not belong to any class
End if
```

**Fig. 4.** Algorithm for computing the response of Braun's cathode machine

In the next step the algorithm maps the set of points on the class number which was assigned to this coordinates in the process of training. If the aforementioned mapping is not possible, algorithm signalizes a failed attempt of classification of input vector.

## 4 Testing Braun's Machine

In this section we will present the results of the tests conducted on Braun's machine. These tests aimed at comparison of the presented machine model with well-known algorithms that are use with several basic notions connected with the classification of multidimensional data. The test task was divided into two categories. The first category is a classification where Braun's machine and selected comparative algorithm are trained to which class a given input vector should be classified during the process of training. During the training phase, algorithms classify data that belong to a testing set and undergo distortion process. The second type of classification is self-organization of data.

### 4.1 Classification

The first collection the Thyroid Disease is associated with medicine and the diagnosis of disease associated with thyroid. The other two data sets - the Landsat Satellite data and the Letter, they are related to image recognition. These data sets were taken from generally accessible data repository UCI Machine Learning Repository [1]. Detailed information about these data sets have been included [1],[2],[3].

Table 1 presents the results of the classification of these data sets using the Braun's machine and other algorithms. The results of other algorithms are derived from the work of [2],[3]. The table shows that, the effectiveness of the Braun's machine can be compared to the best algorithms presented in the table. The relatively bad results, the Braun's machine obtained for the Thyroid data set. In this case, it should be noted that

**Table 1.** The percentage of correct classifications for the data sets. BM–Braun's machine with *L*–dimensional burned surface.(***$U_a/U_d$*** –quotient of the value of accelerating voltage ***$U_a$*** and deflecting voltage ***$U_d$*** *)*

**Thyroid Disease Data Set**

| *Algorithm* | *%Test* |
|---|---|
| CART tree | 99.36 |
| SSV tree | 99.33 |
| MLP+SCG, 4 neurons | 99.24 |
| SVM Minkovsky kernel | 99.18 |
| MLP+SCG, 4 neurons, 45 SV | 98.92 |
| FSM 10 rules | 98.90 |
| MLP+SCG, 12 neurons | 98.83 |
| MLP+backprop | 98.5 |
| SVM Gaussian kernel | 98.4 |
| k-NN, k=1, 8 features | 97.3 |
| **BM *$L$=19, $U_a$/ $U_d$ =16*** | **96.93** |
| Naive Bayes | 96.1 |
| SVM Gauss, C=1 s=0.1 | 94.7 |
| BP+conj. gradient | 93.8 |
| 1-NN Manhattan, | 93.8 |
| SVM lin, C=1 | 93.3 |

**Landsat Satellite Data set**

| *Algorithm* | *%Test* |
|---|---|
| MLP, 36 nodes, +SVNT | 91.3 |
| MLP, 36 nodes, | 91.0 |
| kNN, k=3, Manhattan | 90.9 |
| **BM *$L$=33, $U_a/U_d$ =9*** | **90.7** |
| FSMneurofuzzy,learn0.95 | 89.7 |
| kNN, k=1, Euclidean | 89.4 |
| SVM Gaussian kernel | 88.4 |
| RBF, Statlog result | 87.9 |
| **BM *$L$=31, $U_a$/ $U_d$ =10*** | **87.8** |
| MLP, Statlog result | 86.1 |
| Bayesian Tree | 85.3 |
| C4.5 tree | 85.0 |
| SSV tree | 84.3 |
| Cascade | 83.7 |
| Kohonen | 82.1 |
| Bayes | 71.3 |

**Letter Recognition Data Set**

| *Algorithm* | *%Test* |
|---|---|
| **BM *$L$=14, $U_a/U_d$=1*** | **96.06** |
| **BM *$L$=10, $U_a/U_d$=10*** | **95.18** |
| ALLOC80 | 93.60 |
| K-NN | 93.20 |
| LVQ | 92.10 |
| Quadisc | 88.70 |
| Bayesian Tree | 87.60 |
| NewId | 87.20 |
| IndCART | 87.00 |
| C4.5 | 86.80 |
| **BM *$L$=8, $U_a$/ $U_d$ =10*** | **86.56** |
| DIPOL92 | 82.40 |
| RBF | 76.60 |
| Logdisc | 76.60 |
| Kohonen | 74.80 |
| Backprop | 67.30 |

the number of correct classification obtained using the machine, it is worse by only two percent of the best result. In contrast, the classification of other data sets, Braun's machine, was extremely effective in comparison with other algorithms. For the Letter data set, the machine has achieved the best result. In this case, the difference between the result obtained by the machine, and the third result, is more than two percent. The results in Table 1 show that the effectiveness of the machine is dependent on dimensional burned surface, value of accelerating voltage $\boldsymbol{U_a}$ and deflecting voltage $\boldsymbol{U_d}$. In the case of the data set the letter, an increase of dimensional burned surface, has increased the efficiency of the machine by ten percent. The reason for such a phenomenon is the fact that the dimension of the burned surface increases separation between individual data classes.

### 4.2 Self-organization of Data

In this paragraph we will present the results of the tests of Braun's machine connected with the notion of self-organization of data. The results of the computations conducted will be compare with the results of two selected algorithms. For this type of classification, a set of training and test data were synthetic. A training set is a set of 127-element vectors (9) which belong to forty classes. Testing set consisted of 4000 elements of a given degree of noise.

$$DataSet\ := \bigcup_{i=1}^{10} \cdot \{ i \cdot \sin(\ i \cdot t) \cup (-i \cdot \sin(\ i \cdot t)) \cup i \cdot \cos(\ i \cdot t) \cup (-i \cdot \cos(\ i \cdot t)) \} \qquad (9)$$

There was a difference in comparison to previous example which referred to the process of training. In this case, for a given input vector first an algorithm computing the response of a machine was activated as seen in Fig.5. If this algorithm did not classify a given input vector in any already existing classes , then for this input vector a machine training process was initiated, which we have presented in Fig. 4. Basic criteria applied in the evaluation of particular algorithms were the number of correct classifications during testing and the time of training of a given algorithm.

Table 2. presents the results of the conducted tests. The results of these tests authorize us to make some conclusions. First of all, presented results indicate that Braun's machine showed the highest effectiveness for a small level of test data noise. The percentage of correct responses of this algorithm was proportionally dependent on the dimension of the burned surface. The reason for such a phenomenon is the fact

**Table 2.** The percentage of correct classifications for a test set. $k_{100}$ – Kohonen network with a layer of 100 neurons; $k_{250}$ – Kohonen network with a layer of 250 neurons; $IS_{(optimal\ setting)}$ – immune system; $BM_L$ – Braun's machine with $L$-dimensional burned surface.

| | | | | ***$U_a/U_d$=6*** | | | | ***$U_a/U_d$=10*** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ***Noise*** | ***$K_{100}$*** | ***$K_{250}$*** | ***IS*** | ***$BM_{L=5}$*** | ***$BM_{L=10}$*** | ***$BM_{L=14}$*** | ***$BM_{L=20}$*** | ***$BM_{L=10}$*** | ***$BM_{L=14}$*** | ***$BM_{L=20}$*** |
| ***30%*** | 64 | 70 | 88 | 68 | 97 | 99 | 98 | 92 | 100 | 100 |
| ***40%*** | 55 | 60 | 88 | 57 | 94 | 94 | 96 | 90 | 99 | 100 |
| ***50%*** | 52 | 58 | 88 | 50 | 93 | 88 | 90 | 87 | 98 | 100 |
| ***60%*** | 46 | 54 | 80 | 43 | 77 | 79 | 80 | 81 | 96 | 98 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ***$No_{Class}$*** | 32 | 33 | 39 | 35 | 40 | 40 | 40 | 37 | 40 | 40 |
| ***$t_{train}$*** | 280s | 600s | 480s | 8s | 10s | 11s | 11s | 10s | 11s | 11s |
| ***$No_{Iter}$*** | 200 | 200 | 200 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

that the dimension of the burned surface increases separation between individual data classes. In case of Braun's machine, however, the ability to generalize can by improved be the increase of accelerating voltage. The increase of the value of this parameter results in the increase of correct classifications for the most distorted data, which results from the principles of machine operation. The last three verses of the table being discussed here containing information which refers to a division of a training set into classes, time of training of the tested algorithms and the number of iteration in case of each algorithms during the training phase. Table 2. shows that only Braun's machine was able to "discover" all classes belonging to a training set. The effectiveness of this division was determined by two factors. First of all, it was connected with the dimension of the burned surface. Secondly, as it can be derived from table 1, the effectiveness of this division also depended on $U_a$ parameter which is the counterpart of accelerating voltage. It can be noticed that the time of Braun's machine training is at least by one order of magnitude shorter that the time other algorithms need for training. Such huge differences in size result from the fact that Braun's machine training algorithm is a one-phase algorithm (see Fig. 3) and it is connected with a single computation of activation function for a given data vector. In case of other algorithms, however, the process of training is an iterative process.

## 5 Summary

In this article we have presented a model used for a classification of multidimensional data in a broader sense, called Braun's machine. Functioning of this machine has been based on principles which govern the particles in motion in electric field. The internal structure of the machine presented ion this paper has been based on the architecture of a cathode-ray tube – Braun's tube. For a machine model described this way a Braun's machine training algorithm has been proposed as well as response computing algorithms. In the final chapter we have presented the results of the machine tests for the notions connected with the classification and self-organization of multidimensional data. The results we have turned out to be very interesting, particularly when it terms of the notions connected with self-organization. During this type of tests it appeared that the responses generated by Braun's machine were equally good as comparative algorithms, or even in some case their quality was better. On the basis of the conducted research it can be said that the machine model is a very interesting computational model due to its limited computational complexity.

## References

1. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases, University of California, Irvine, Dept. of Information and Computer Science (1998-2003), `http://www.ics.uci.edu/mlearn/MLRepository.html`
2. Duch, W.: Support Vector Neural Training. IEEE Transactions on Neural Networks (2004)
3. Statlog Datasets comparison of results: Department of Informatics Nicolaus Copernicus University (2009), `http://www.is.umk.pl/projects/datasets.html`
4. Tipler, P.: Physics for Scientists and Engineers: Electricity, Magnetism, Light, and Elementary Modern Phsics, 5th edn. W.H. Freeman, New York (2004)

# Fuzzy Decision Tree Induction Approach for Mining Fuzzy Association Rules

Rolly Intan and Oviliani Yenty Yuliana

Informatics Engineering Department
Petra Christian University, Surabaya, Indonesia
rintan@peter.petra.ac.id, ovi@peter.petra.ac.id

**Abstract.** Decision Tree Induction (DTI), one of the Data Mining classification methods, is used in this research for predictive problem solving in analyzing patient medical track records. In this paper, we extend the concept of DTI dealing with meaningful fuzzy labels in order to express human knowledge for mining fuzzy association rules. Meaningful fuzzy labels (using fuzzy sets) can be defined for each domain data. For example, fuzzy labels *poor disease*, *moderate disease*, and *severe disease* are defined to describe a condition/type of disease. We extend and propose a concept of fuzzy information gain to employ the highest information gain for splitting a node. In the process of generating fuzzy association rules, we propose some fuzzy measures to calculate their support, confidence and correlation. The designed application gives a significant contribution to assist decision maker for analyzing and anticipating disease epidemic in a certain area.

**Keywords:** Data Mining, Classification, Decision Tree Induction, Fuzzy Set, Fuzzy Association Rules.

## 1 Introduction

Decision Tree Induction (DTI) has been used in machine learning and in data mining as a model for prediction a target value based on a given relational database. There are some commercial decision tree applications, such as the application for analyzing a return payment of a loan for owning or renting a house [15] and the application of software quality classification based on the program modules risk [16]. Both applications inspire this research to develop an application for analyzing patient medical track record. The Application is able to present relation among (single/group) values of patient attribute in decision tree diagram. In the developed application, some domains of data need to be utilized by meaningful fuzzy labels. For example, fuzzy labels *poor disease*, *moderate disease*, and *severe disease* describe a condition/type of disease; *young*, *middle aged* and *old* are used as the fuzzy labels of ages. Here, a fuzzy set is defined to express a meaningful fuzzy label. In order to utilize the meaningful fuzzy labels, we need to extend the concept of (*crisp*) DTI using fuzzy approach. Simply, the extended concept is called *Fuzzy Decision Tree* (FDT). To generate FDT from a normalized database that consists of several tables, there are several

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 720–728, 2009.

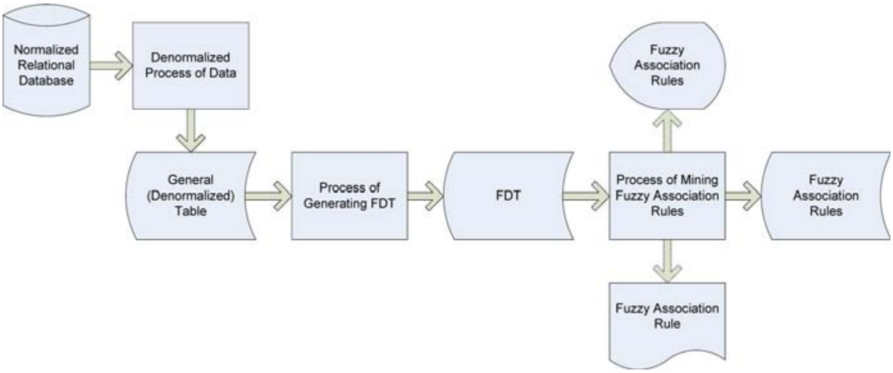

Normalized Relational Database
Denormalized Process of Data
General (Denormalized) Table
Process of Generating FDT
FDT
Process of Mining Fuzzy Association Rules
Fuzzy Association Rules
Fuzzy Association Rules
Fuzzy Association Rule

**Fig. 1.** Process of Mining Association Rules

sequential processes as shown in Fig. 1. First is the process of joining tables known as *Denormalization of Database* as discussed in [4]. The process of denormalization can be provided based on the relation of tables as presented in Entity Relationship Diagram (ERD) of a relational database. Result of this process is a general (denormalized) table. Second is the process of constructing FDT generated from the denormalized table.

In the process of constructing FDT, we propose a method how to calculate fuzzy information gain by extending the existed concept of (*crisp*) information gain to employ the highest information gain for splitting a node. The last is the process of mining fuzzy association rules. In this process, fuzzy association rules are mined from FDT. In the process of mining fuzzy association rules, we propose some fuzzy measures to calculate their support, confidence and correlation. Minimum support, confidence and correlation can be given to reduce the number of mining fuzzy association rules. The designed application gives a significant contribution to assist decision maker for analyzing and anticipating disease epidemic in a certain area.

The structure of the paper is the following. Section 2 as main contribution of this paper is devoted to propose the concept and algorithm for generating FDT. Section 3 proposes some equations of fuzzy measures that play important role in the process of mining fuzzy association rules. Section 4 demonstrates the algorithm and in a simple illustrative results. Finally a conclusion is given in Section 5.

## 2 Fuzzy Decision Tree Induction (FDT)

Based on type of data, we may classify DTI into two types, namely crisp and fuzzy DTI. Both DTI are compared based on Generalization-Capability [14]. The result shows that Fuzzy Decision Tree (FDT) is better than Crisp Decision Tree (CDT) in providing numeric attribute classification. Fuzzy Decision Tree formed by the FID3, combined with Fuzzy Clustering (to form a function member) and validated cluster (to decide granularity) is also better than Pruned Decision Tree. Here, Pruned

Decision Tree is considered as a Crisp enhancement [13]. Therefore in our research work, disease track record analyzer application development, we propose a kind of FDT using fuzzy approach.

An information gain measure [1] is used in this research to select the test attribute at each node in the tree. Such a measure is referred to as an attribute selection measure or a measure of the goodness of split. The attribute with the highest information gain (or greatest entropy reduction) is chosen as the test attribute for the current node. This attribute minimizes the information needed to classify the samples in the resulting partitions and reflects the least randomness or impurity in these partitions. In order to process crisp data, the concept of information gain measure is defined in [1] by the following definitions.

Let S be a set consisting of *s* data samples. Suppose the class label attribute has *m* distinct values defining *m* distinct classes, $C_i$ (for *i*=1,…, *m*). Let $s_i$ be the number of samples of S in class $C_i$. The expected information needed to classify a given sample is given by

$$I(s_1, s_2, ..., s_m) = -\sum_{i=1}^{m} p_i \log_2(p_i) \tag{1}$$

where $p_i$ is the probability that an arbitrary sample belongs to class $C_i$ and is estimated by $s_i/s$.

Let attribute *A* have *v* distinct values, $\{a_1, a_2, ..., a_v\}$. Attribute *A* can be used to partition *S* into *v* subsets, $\{S_1, S_2, ..., S_v\}$, where $S_j$ contains those samples in *S* that have value $a_j$ of *A*. If *A* was selected as the test attribute then these subsets would correspond to the braches grown from the node containing the set *S*. Let $s_{ij}$ be the number of samples of class $C_i$ in a subset $S_j$. The entropy, or expected information based on the partitioning into subsets by *A*, is given by

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + ... + s_{mj}}{s} I(s_{1j}, ..., s_{mj}) \tag{2}$$

The term $\frac{s_{ij} + ... + s_{mj}}{s}$ acts as the weight of the *j*th subset and is the number of samples in the subset divided by the total number of samples in *S*. The smaller the entropy value, the greater the purity of the subset partitions.The encoding information that would be gained by branching on A is

$$Gain(A)=I(s_1, s_2,..., s_m) - E(A) \tag{3}$$

In other words, *Gain(A)* is the expected reduction in entropy caused by knowing the values of attribute *A*.

When using the fuzzy value, the concept of information gain as defined in (1) to (3) will be extended to the following concept. Let S be a set consisting of *s* data samples. Suppose the class label attribute has *m* distinct values, $v_i$ (for *i*=1,…, *m*), defining *m* distinct classes, $C_i$ (for *i*=1,…, *m*). And also suppose there are *n* meaningful fuzzy labels, $F_j$ (for *j*=1,…, *n*) defined on *m* distinct values, $v_i$. $F_j(v_i)$ denotes membership degree of $v_i$ in the fuzzy set $F_j$ . Here, $F_j$ (for $j$=1,…, $n$) is defined by satisfying the following property:

$$\sum_{j}^{n} F_j(v_i) = 1, \forall \mathrm{i} \in \{1,...m\}$$

Let $\beta_j$ be a weighted sample corresponding to $F_j$ as given by

$\beta_j = \sum_{i}^{m} \det(C_i) \times F_j(v_i)$ , where det($C_i$) is the number of elements in $C_i$.

The expected information needed to classify a given weighted sample is given by

$$I(\beta_1, \beta_2, ..., \beta_n) = -\sum_{j=1}^{n} p_j \log_2(p_j) \tag{4}$$

where $p_j$ is estimated by $\beta_j$/s.

Let attribute $A$ have $u$ distinct values, $\{a_1, a_2, ..., a_u\}$, defining $u$ distinct classes, $B_h$ (for $h$=1,…, $u$). Suppose there are $r$ meaningful fuzzy labels, $T_k$ (for $k$=1,…, $r$), defined on $A$. Similarly, $T_k$ is also satisfy the following property.

$$\sum_{k}^{r} T_k(a_h) = 1, \forall h \in \{1,...,u\}$$

If $A$ was selected as the test attribute then these fuzzy subsets would correspond to the braches grown from the node containing the set $S$. The entropy, or expected information based on the partitioning into subsets by $A$, is given by

$$E(A) = \sum_{k=1}^{r} \frac{\alpha_{1k} + ... + \alpha_{nk}}{s} I(\alpha_{1k}, ..., \alpha_{nk}) \tag{5}$$

Where $\alpha_{jk}$ be intersection between $F_j$ and $T_k$ defined on data sample $S$ as follows.

$$\alpha_{jk} = \sum_{h}^{u} \sum_{i}^{m} \min(F_j(v_i), T_k(a_h)) \times \det(C_i \cap B_h) \tag{6}$$

Similar to (4), $I(\alpha_{ik}, ..., \alpha_{nk})$ is defined as follows.

$$I(\alpha_{1k}, ..., \alpha_{nk}) = -\sum_{j=1}^{n} p_{jk} \log_2(p_{jk}) \tag{7}$$

where $p_{jk}$ is estimated by $\alpha_{jk}$/s.

Finally, the encoding information that would be gained by branching on A is

$$Gain(A) = I(\beta_1, \beta_2, ..., \beta_n) - E(A) \tag{8}$$

Since fuzzy sets are considered as a generalization of crisp set, it can be proved that the equations (4) to (8) are also generalization of equations (1) to (3).

## 3 Mining Fuzzy Association Rules from FDT

*Association rules* are kind of patterns representing correlation of attribute-value (items) in a given set of data provided by a process of data mining system. Generally, association rule is a conditional statement (such kind of *if-then rule*). Performance or interestingness of an association rule is generally determined by three factors, namely *confidence*, *support* and *correlation* factors. Confidence is a measure of certainty to assess the validity of the rule. The support of an association rule refers to the percentage of relevant data tuples (or transactions) for which the pattern of the rule is true. Correlation factor is another kind of measures to evaluate correlation between two entities.

Related to the proposed concept of FDT as discussed in Section 2, the fuzzy association rule, $T_k \Rightarrow F_j$ can be generated from the FDT. The confidence, support and correlation of $T_k \Rightarrow F_j$ are given by

$$\text{confidence}(T_k \Rightarrow F_j) = \frac{\sum_h^u \sum_i^m \min(F_j(v_i), T_k(a_h)) \times \det(C_i \cap B_h)}{\sum_h^u T_k(a_h) \times \det(B_h)} \tag{9}$$

$$\text{support}(T_k \Rightarrow F_j) = \frac{\sum_h^u \sum_i^m \min(F_j(v_i), T_k(a_h)) \times \det(C_i \cap B_h)}{s} \tag{10}$$

$$\text{correlation}(T_k \Rightarrow F_j) = \frac{\sum_h^u \sum_i^m \min(F_j(v_i), T_k(a_h)) \times \det(C_i \cap B_h)}{\sum_h^u \sum_i^m F_j(v_i) \times T_k(a_h) \times \det(C_i \cap B_h)} \tag{11}$$

To provide a more generalized multidimensional fuzzy association rules as proposed in [6], it is started from a single table (relation) as a source of data representing relation among item data. Formally, a relational data table [12] $R$ consists of a set of tuples, where $t_i$ represents the $i$-th tuple and if there are $n$ domain attributes $D$, then $t_i = (d_{i1}, d_{i2}, \cdots, d_{in})$. Here, $d_{ij}$ is an atomic value of tuple $t_i$ with the restriction to the domain $D_j$, where $d_{ij} \in D_j$. A relational data table $R$ is defined as a subset of the set of cross product $D_1 \times D_2 \times \cdots \times D_n$, where $D = \{D_1, D_2, \cdots, D_n\}$. Tuple $t$ (with respect to $R$) is an element of $R$. In general, $R$ can be shown in Table 1.

Now, we consider $\chi$ and $\psi$ as subsets of fuzzy labels. Simply, $\chi$ and $\psi$ are called fuzzy datasets. A fuzzy dataset is a set of fuzzy data consisting of several distinct fuzzy labels, where each fuzzy label is represented by a fuzzy set on a certain domain attribute. Formally, $\chi$ and $\psi$ are given by $\chi = \{F_j \mid F_j \in \Omega(D_j),\ \exists\ j \in \mathrm{N}_n\}$ and

**Table 1.** A Schema of Relational Data Table

| Tuples | $D_1$ | $D_2$ | $\dots$ | $D_n$ |
|---|---|---|---|---|
| $t_1$ | $d_{11}$ | $d_{12}$ | $\dots$ | $d_{1n}$ |
| $t_2$ | $d_{21}$ | $d_{22}$ | $\dots$ | $d_{2n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| $t_s$ | $d_{s1}$ | $d_{s2}$ | $\dots$ | $d_{sn}$ |

$\psi = \{F_j \mid F_j \in \Omega(Dj),\ \exists\ j \in \mathrm{N}_n\}$, where there are $n$ domain data, and $\Omega(D_j)$ is a fuzzy power set of $D_j$. In other words, $F_j$ is a fuzzy set on $D_j$. The confidence, support and correlation of $\chi \Rightarrow \psi$ are given by

$$\mathrm{support}(\chi \Rightarrow \psi) = \frac{\sum_{i=1}^{s} \inf_{F_j \in \chi \cup \psi} \{F_j(d_{ij})\}}{s} \tag{12}$$

$$\mathrm{confidence}(\chi \Rightarrow \psi) = \frac{\sum_{i=1}^{s} \inf_{F_j \in \chi \cup \psi} \{F_j(d_{ij})\}}{\sum_{i=1}^{s} \inf_{F_j \in \chi} \{F_j(d_{ij})\}} \tag{13}$$

$$\mathrm{correlation}(\chi \Rightarrow \psi) = \frac{\sum_{i=1}^{s} \inf_{F_j \in \chi \cup \psi} \{F_j(d_{ij})\}}{\sum_{i=1}^{s} \inf_{A_j \in \chi} \{A_j(d_{ij})\} \times \inf_{B_k \in \psi} \{B_k(d_{ik})\}} \tag{14}$$

## 4 FDT Algorithms and Results

The research is conducted based on the Software Development Life cycle method. The application design conceptual framework is shown in Fig 1. An input for developed application is a single table that is produced by denormalization process from a relational database. The main algorithm for mining association rule process, i.e. Decision Tree Induction, is shown in Fig 2. Furthermore, the procedure for calculating information gain, to implementing equation (4), (5), (6), (7) and (8), is shown in Fig 3. Based on the highest information gain the application can develop decision tree in which the user can display or print it. The rules can be generated from the generated decision tree. Equation (9), (10) and (11) are used to calculate the interestingness or performance of every rule. The number of rules can be reduced based on their degree of support, confidence and correlation compared to the minimum value of support, confidence and correlation determined by user.

```
 For i=0 to the total level
    Check whether the level had already split
    If the level has not yet split Then
      Check whether the level can still be split
      If the level can still be split Then
      Call the procedure to calculate information gain
        Select a field with the highest information gain
        Get a distinct value of the selected field
        Check the total distinct value
        If the distinct value is equal to one Then
        Create a node with a label from the value name
        Else
        Check the total fields that are potential to become a current test attribute
        If no field can be a current test attribute Then
           Create a node with label from the majority value name
        Else
           Create a node with label from the selected value name
        End If
      End If
  End If
End If
 End for
 Save the input create tree activity into database
```

**Fig. 2.** The Generating Decision Tree Algorithm

```
Calculate gain for a field as a root
Count the number of distinct value field
For i=0 to the number of distinct value field
    Count the number of distinct value root field
    For j=0 to the number of distinct value root field
        Calculate the gain field using equation (4) and (8)
    End For
    Calculate entropy field using equation (5)
End For
Calculate information gain field
```

**Fig. 3.** The Procedure to Calculate Information Gain

In this research, we implement two data types as a fuzzy set, namely alphanumeric and numeric. An example of alphanumeric data type is *disease*. We can define some meaningful fuzzy labels of *disease*, such as *poor disease*, *moderate disease*, and *severe disease*. Every fuzzy label is represented by a given fuzzy set. The *age* of patients is an example of numeric data type. *Age* may have some meaningful fuzzy labels such as *young* and *old*. Fig 4 shows an example result of FDT applied into three domains (attributes) data, namely *Death*, *Age* and *Disease*.

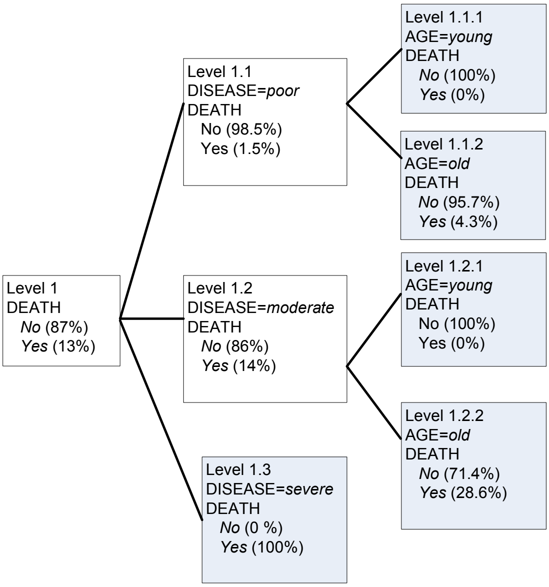


**Fig. 4.** The Generated Decision Tree

## 5 Conclusion

The paper discussed and proposed a method to extend the concept of Decision Tree Induction using fuzzy value. Some generalized formulas to calculate information gain ware introduced. In the process of mining fuzzy association rules, some equations ware proposed to calculate support, confidence and correlation of a given association rules. Finally, an algorithm was briefly given to show the process how to generate FDT.

## Acknowledgment

This research was supported by research grant PHB (244/SP2H/PP/DP2M/III/2008) and HIKOM (25/SP2H/PP/DP2M/V/2009) from Indonesian Higher Education Directorate.

## References

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. The Morgan Kaufmann Series (2001)
2. Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall, New Jersey (1995)
3. Intan, R.: An Algorithm for Generating Single Dimensional Association Rules. Jurnal Informatika (Terakreditasi SK DIKTI No. 56/DIKTI/Kep/2005) 7(1) (May 2006)

4. Intan, R.: A Proposal of Fuzzy Multidimensional Association Rules. Jurnal Informatika (Terakreditasi SK DIKTI No. 56/DIKTI/Kep/2005) 7(2) (November 2006)
5. Intan, R.: A Proposal of an Algorithm for Generating Fuzzy Association Rule Mining in Market Basket Analysis. In: Proceeding of CIRAS (IEEE). Singapore (2005)
6. Intan, R.: Generating Multi Dimensional Association Rules Implying Fuzzy Valuse. In: The International Multi-Conference of Engineers and Computer Scientist, Hong Kong (2006)
7. Gunawan, O.P.: Perancangan dan Pembuatan Aplikasi Data Mining dengan Konsep Fuzzy c-Covering untuk Membantu Analisis Market Basket pada Swalayan X (in Indonesian) Final Project (2004)
8. Zadeh, L.A.: Fuzzy Sets and systems. International Journal of General Systems 17, 129–138 (1990)
9. Agrawal, R., Imielimski, T., Swami, A.N.: Mining Association Rules between Sets of Items in Large Database. In: Proccedings of ACM SIGMOD International Conference Management of Data, pp. 207–216. ACM Press, New York (1993)
10. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: Proccedings of 20th International Conference Very Large Database, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
11. Codd, E.F.: A Relational Model of Data for Large Shared Data Bank. Communication of the ACM 13(6), 377–387 (1970)
12. Benbrahim, H., Amine, B.: A Comparative Study of Pruned Decision Trees and Fuzzy Decision Trees. In: Proceedings of 19th International Conference of the North American, Atlanta, pp. 227–231 (2000)
13. So, Y.D., Sun, J., Wang, X.Z.: An Initial comparison of Generalization-Capability between Crisp and fuzzy Decision Trees. In: Proceedings of the First International Conference on Machine Learning and Cybernetics, pp. 1846–1851 (2002)
14. ALICE d'ISoft v.6.0 demonstration, http://www.alice-soft.com/demo/al6demo.htm (Accessed: 31 October 2007)
15. Khoshgoftaar Taghi, M., Liu, Y., Seliya, N.: Genetic Programming-Based Decision Trees for Software Quality Classification. In: Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, California, pp. 374–383 (2003)

# AdaIndex: An Adaptive Index Structure for Fast Similarity Search in Metric Spaces

Tao Ban[1], Shanqing Guo[2], Qiuliang Xu[2], and Youki Kadobayashi[1]

[1] National Institute of Information and Communications Technology, Tokyo, 184-8795 Japan
[2] School of Computer Science and Technology, Shandong University, Jinan, China

**Abstract.** The Distance Index (D-index) is a recently introduced metric indexing structure which has state-of-the-art performance in large scale metric search applications. Inspired by D-index, we introduce a novel index structure, termed AdaIndex, for fast similarity search in generic metric spaces. With multiple principles from other advanced algorithms, AdaIndex shows a significant improvement in reduction of distance calculations compared with D-index. To treat with application with different system limitations and diverse nature of data, we introduce a parameter tuning algorithm to build an optimal AdaIndex structure with minimal overall computational costs. The efficiency of AdaIndex is validated on a series of simulation experiments.

## 1 Introduction

Similarity search [1,2] has always been a prominent data preprocessing operation in fields like data mining, signal processing, multimedia information retrieval, computational biology, pattern recognition, etc. With the increasing complexity of modern data types, metric spaces have become a popular paradigm for similarity search in applications where no information beyond the inter-object distances is available.

Similarity hashing methods known as Distance Index (D-index) [3] and its descendants incorporate multiple principles for advanced search efficiency. D-index organizes objects into a hierarchical composed of multiple *levels*, where an individual level is further hashed into *separable buckets* by some *split function*. At query time, the levels are sequentially accessed. Because objects in difference buckets are search-separable up to some predefined value $\rho$ and at most one bucket per level needs to be accessed for queries with search range $r \leq \rho$, D-index supports a bounded search cost. According to the experiments in [3], D-index showed promising results in terms of reduction of distance calculations compared with other popular metric index methods. Moreover, by storing the compact clusters of objects in the separable buckets into disk pages, it also offered an excellent IO management performance. D-index has built a good framework for metric search especially for queries with comparatively small radii. However, the pivot selection strategy suggested in [3] does not support a balanced data partition in the structure and thus deteriorates the search performance and increases the IO cost at query time. Another problem of D-index is the tuning of parameter $\rho$: With a large $\rho$ value, there will be too many objects assigned into the inseparable regions, on the other hand, a small $\rho$ value always indicates degenerated search separable property for the separable buckets.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 729–737, 2009.

In this paper, we present an adaptive indexing structure termed AdaIndex, which efficiently addresses the above two problems of D-index. When partitioning the objects into separable sets, D-index fixes the $\rho$ value and leaves the formulation of the separable sets to a pivot selection procedure. On the contrary, in AdaIndex, we fix the size of the separable sets and try to maximize the search-separable property between them. The alternated strategy will lead to the following benefits. First, the number of pivots no longer influences the formulation of the separate sets so that we gain degrees of freedom to tune the number of pivots in the index structure. We can either select more pivots to improve the search performance or reduce the number to implement the index structure on a system with limited storage resource. Second, the maximization of the search-separable property will result in better search performance, substantially reducing the side computations at query time. Finally, with the objects evenly distributed into search separable sets, more efficient IO management can be supported by the algorithm.

## 2 Metric Search by AdaIndex

The proposed AdaIndex is designed for similarity search in generic metric spaces.

### 2.1 Partitioning the Dataset

In AdaIndex, we use the split functions to hash objects into search-separable clusters. A *bps* (ball-partitioning split) function is defined by a tuple $(\boldsymbol{p}_m, K)$, where $\boldsymbol{p}_m$ is a pivot, i.e., a selected anchor object, and $K$ the size of the neighboring set to the pivot. Let the distance from $\boldsymbol{p}_m$ to its $K$th nearest neighbor be $d_K$. Then a *bps* uniquely determines the belongingness of any object $\boldsymbol{x}_i$ in the indexed dataset $\boldsymbol{X}$:

$$bps^1(\boldsymbol{x}_i) = \begin{cases} 1 & \text{if } d(\boldsymbol{x}_i, \boldsymbol{p}_m) \le d_K, \\ 0 & \text{if } d(\boldsymbol{x}_i, \boldsymbol{p}_m) > d_K. \end{cases} \tag{1}$$

The superscript 1 denotes the order of the split function, i.e., the number of pivots involved. To partition the dataset into more subsets, we compose higher order $\rho$-split functions by combining multiple first order *bps* functions. Given $M$ *bps* functions, the joint $M$-order split function is denoted as $bps^M$, and the return value can be seen as a string $b = [b_1, b_2, \cdots, b_m, \cdots, b_M]$, $b_m \in \{0, 1\}$. The following $\langle\cdot\rangle$ operator returns an integer value in $[0, M]$ for any string $b$:

$$\langle b \rangle = \begin{cases} m, & \text{if } \exists m \;\; b_m = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

We call the neighboring subset associated with a pivot the *separable set* and the objects which do not belong to any of the separable sets are assigned to an *exclusion set*. If the pivots are selected to be far from each other, then we can achieve a better separable ability for the separable sets. Fig. 1a shows an example where a collection of objects are divided into three separable sets, $\boldsymbol{S}_1$, $\boldsymbol{S}_2$, $\boldsymbol{S}_3$, and an exclusion set $\boldsymbol{E}$.

To make sure no object will be assigned to multiple subsets, we impose an ordering among the pivots – the neighbor of the previously selected pivots are removed before a subsequent pivot takes effect.

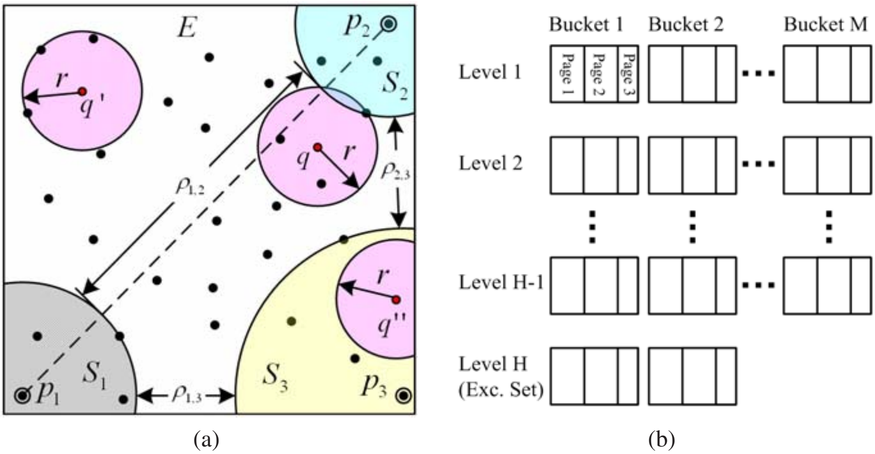


**Fig. 1.** AdaIndex data structure. (a) Single level partition on a 2D dataset, $K = 3$, $M = 3$. (b) The multilevel storage structure of AdaIndex.

### 2.2 Search Separability

In Fig. 1a, we call the minimal distance between two separable sets a *margin*, noted as $\rho_{i,j}$. The feature that two separable sets satisfies

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) > \rho_{i,j}, \qquad \forall \boldsymbol{x}_1 \in \boldsymbol{S}_i,\ \boldsymbol{x}_2 \in \boldsymbol{S}_j. \tag{3}$$

is called the *search separable property*. This property indicates that if the search radius $r \leq \rho_{i,j}$, then for an arbitrary query object $\boldsymbol{q}$ only one of $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$ needs to be accessed. For example, for query $\boldsymbol{q}$ in Fig. 1a, the query sphere intersects $\boldsymbol{S}_2$, and because of the search separable property, objects in $\boldsymbol{S}_1$ and $\boldsymbol{S}_3$ can be safely pruned. Of course, there is some chance that the query sphere falls exclusively into the exclusion set and thus no separable set needs to be accessed in the level, e.g., query $\boldsymbol{q}'$ in Fig. 1a. Obviously, the margins among separable sets give a good measure on the search difficulty in the level and need to be maximized for better search performance.

### 2.3 Multilevel Structure

Naturally, the margins of the separable sets diminish as more pivots are added. A simple but effective approach to obtaining large margins between separable sets is to formulate a hierarchical structure. The data structure associated with the separable sets defined by an $M$-order *bps* function is called a *level*. Note that because the objects in the exclusion set do not satisfy the search separable property so that group pruning is not supported. Moreover, as the objects in the exclusion set can not be organized into geometrically compact groups, the random access of the objects will degenerate the IO performance of the algorithm. Hence when the exclusion set of a level is still too large, we create an additional level structure with new split functions applied to form more separable sets. This process is repeated until the exclusion set is empty. Though the last level

**Table 1.** Pivot selection and separable set creation on a level

```
0  Inputs: $\boldsymbol{X}$, $M$, $K$;
1  $\boldsymbol{P} \leftarrow \phi$; $\boldsymbol{S}_m \leftarrow \phi, m = 1, \cdots, M$;
2  randomly select $\boldsymbol{p}_1 \in \boldsymbol{X}$;
3  for $m = 1$ to $M$ do
4      $\boldsymbol{P} \leftarrow \boldsymbol{P} \cup \boldsymbol{p}_m$; $\boldsymbol{X} \leftarrow \boldsymbol{X} \setminus \boldsymbol{p}_m$; $\boldsymbol{S}_m \leftarrow \boldsymbol{S}_m \cup \boldsymbol{p}_m$;
5      $d_K \leftarrow d(\boldsymbol{p}_m, \boldsymbol{x}_K)$, where $\boldsymbol{x}_K$ is the $K$th neighbor of $\boldsymbol{p}_m$ in $\boldsymbol{X}$;
6      for $\boldsymbol{x}_i \in \boldsymbol{X}$ do
7          if $d(\boldsymbol{x}_i, \boldsymbol{p}_m) \leq d_K$ then
8              $\boldsymbol{S}_m \leftarrow S_m \cup \boldsymbol{x}_i$; $\boldsymbol{X} \leftarrow \boldsymbol{X} \setminus \boldsymbol{x}_i$; // assign the $K$NNs to the separable set
9          else
10             $d_i \leftarrow \min_{\boldsymbol{p}_m \in \boldsymbol{P}} d(\boldsymbol{x}_i, \boldsymbol{p}_m)$; // update the distance to the pivot set
11         end if
12     end for
13     $\boldsymbol{p}_m \leftarrow \boldsymbol{x}_j$, where $d(\boldsymbol{x}_j, \boldsymbol{P}) = \max_{\boldsymbol{x}_i \in \boldsymbol{X}} d(\boldsymbol{x}_i, \boldsymbol{P})$; // select next pivot
14 end for
15 return $\boldsymbol{P}, \boldsymbol{S}_m$;
```

usually contains a flexible number of separable sets, in the following, we will not make difference of it from other preceding levels.

### 2.4 Pivot Selection

Another approach to increasing the margin of the separable sets is to incorporate some specific strategy in the pivot selection procedure. As well known is the literature, selection of pivots greatly affects the search performance for any search algorithm. We adopt the so-called *farthest point clustering* [4] to select the set of $M$ pivots, $\boldsymbol{P}$. The pivots can be found following the algorithm specified in Table 1. It starts from the initialization of the pivot set with a pivot randomly selected from $\boldsymbol{X}$ (line 2); and then sequentially selects the remaining $M - 1$ pivots as the objects which are farthest away from the current pivot set (line 13). When a pivot is selected, a separable set is created based on the pivot and is assigned with its $K$ nearest neighbors (line 8).

### 2.5 Storage Architecture

D-index consists of a two dimensional array of so called *buckets* which deal with IO and search operations of objects in the separable sets, as shown in Fig. 1b. In the vertical picture, the bucket lists for all $H$ levels are ordered sequentially. In the horizontal picture at a given level, a list of separable sets are obtained from the $M$-order *bps* function on the exclusion set of the previous level. Each of these separable sets is organized as a bucket. Based on the nature of the data, when necessary, a bucket is implemented as an elastic structure consisting of a necessary number of fixed-size *blocks* (disk pages), which is the basic disk access unit of the database system.

## 2.6 Search Operation

Given an AdaIndex structure, the *search algorithm* guides the search procedure. For brevity, we only discuss the algorithm for range search, which is known as the most basic type of similarity search. As shown in Table 2, the search algorithm sequentially traverses the $H$ levels in the AdaIndex structure. When visiting a level, the algorithm first computes the distances from the query object to the pivots in the level and then applies the following prune rules to all separable sets [5]. Let the maximal and minimal distances between a pivot $\boldsymbol{p}$ and the objects in separable set $\boldsymbol{S}$ be $\delta_{max}$ and $\delta_{min}$, and suppose they are prestored in the index structure. Then, we have

**Rule 1** (Fig. 2a)**:** $\boldsymbol{S}$ can not contain the answer to the query, if $d(\boldsymbol{q}, \boldsymbol{p}) - r > \delta_{max}$;
**Rule 2** (Fig. 2b)**:** $\boldsymbol{S}$ can not contain the answer to the query, if $d(\boldsymbol{q}, \boldsymbol{p}) + r < \delta_{min}$.

If the above two rules fail at a separable set, the set is retrieved from the disk for further examination. To avoid unnecessary distance computations in the set, we make use of another two prune rules [6]. Suppose the distances between an object $\boldsymbol{x}$ in the separable set and a pivot $\boldsymbol{p}$ are prestored in the index structure, then we have

**Rule 3** (Fig. 2c)**:** $\boldsymbol{x}$ is not an answer to the query, if $d(\boldsymbol{x}, \boldsymbol{p}) + r < d(\boldsymbol{q}, \boldsymbol{p})$;
**Rule 4** (Fig. 2d): $\boldsymbol{x}$ is not an answer to the query, if $d(\boldsymbol{x}, \boldsymbol{p}) - r > d(\boldsymbol{q}, \boldsymbol{p})$.

If Rules 3 and 4 fail at an object $\boldsymbol{x}$, then distance $d(\boldsymbol{x}, \boldsymbol{q})$ is directly computed. And when the distance function is invoked, the answer set is updated if $\boldsymbol{x}$ satisfies the query.

Rules 1 and 2 prune multiple objects at a time and thus we call them *group-prune rules*. Rules 3 and 4 are called *object-prune rules* since they prune one object at a time. Obviously, the object-prune rules are the special cases of group-prune rules when there is only one object contained in the separable set. If an object can be pruned by the group-prune rules, then it can be pruned by the object-prune rules as well, given the necessary prestored distances. Thus application of the object-prune rules is the key to the reduction of invoked distance computations at query time. On the other hand, the group-prune rules can substantially avoid unnecessary side computations, i.e., application of the object-prune rules. Moreover, storing $\delta_{max}$ and $\delta_{min}$ is usually much cheaper than holding an array of distances in the main memory. Hence the group-prune rules also contribute to the search performance, especially for applications with computationally less intensive distance functions and tighter storage limitation.

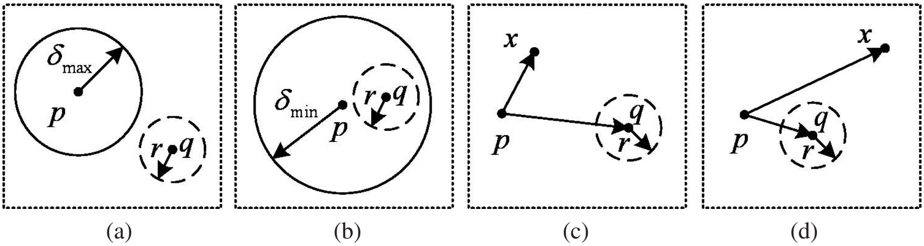


**Fig. 2.** Prune rules. (a) Rule 1. (b) Rule 2. (c) Rule 3. (d) Rule 4.

**Table 2.** Range search algorithm for AdaIndex

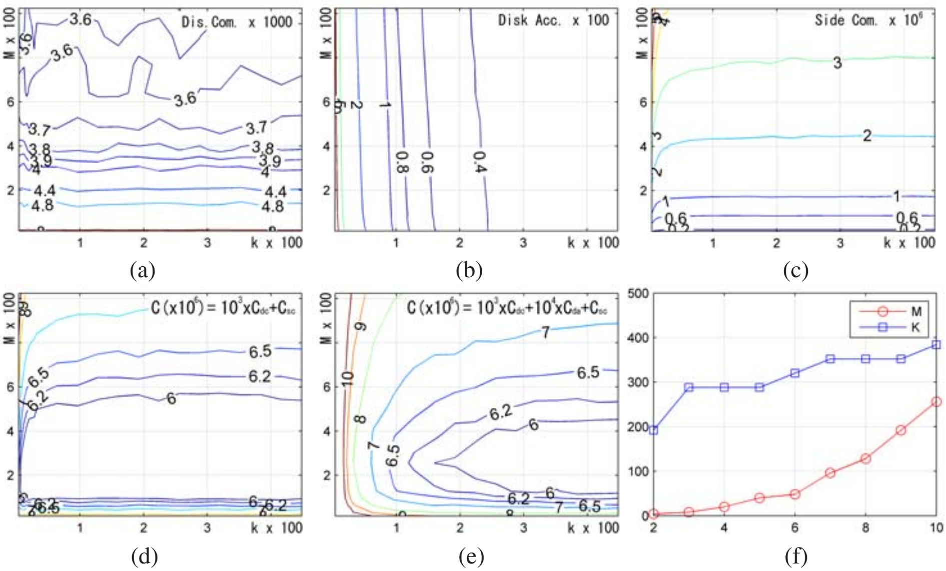
```
Inputs: $\boldsymbol{X}$, $\boldsymbol{q}$, $r$;
$\boldsymbol{A} \leftarrow \emptyset$; // answer set
for $h = 1$ to $H$ do // sequentially traverses all levels
    $\boldsymbol{C} \leftarrow \{S_m, m = 1, \cdots, M\}$; // yet not pruned separable sets
    for $m = 1$ to $M$ do
        $d_m = metric(\boldsymbol{q}, \boldsymbol{p}_m)$; // distance computation
        if $d_m \leq r$ then $\boldsymbol{A} \leftarrow \boldsymbol{A} \cup \boldsymbol{p}_m$; end if // update answer set
        for $l = 1$ to $M$ do // prune separable sets by $\boldsymbol{p}_m$
            if $d_m - r > \delta_{max}[l, m]$ or $d_m + r < \delta_{min}[l, m]$ then // Rule 1, 2
                $\boldsymbol{C} \leftarrow \boldsymbol{C} \setminus S_l$; // separable set pruned
            end if
        end for
    end for
    for all $S_m \subset \boldsymbol{C}$ do // explore the remaining separable sets
        read $\boldsymbol{x}_l (l = 1, \cdots, K)$ from disk; // IO access
        for $m = 1$ to $M$ do // prune individual objects by $\boldsymbol{p}_m$
            for $l = 1$ to $K$ do
                if $d[l, m] + r < d_m$ or $d[l, m] - r > d_m$ then // Rule 3, 4
                else
                    $d'_l = metric(\boldsymbol{x}_l, \boldsymbol{q})$; // distance computation
                    if $d'_l \leq r$ then $\boldsymbol{A} \leftarrow \boldsymbol{A} \cup \boldsymbol{p}_m$; end if // update answer set
                end if
            end for
        end for
    end for
end for
return $\boldsymbol{A}$;
```

## 3 Experiments

In this section, the performance of the index structures is measured by three criteria, i.e., distance calculations, disk accesses, and side computations. In each of the reported experiments, the indexed set, the validation set, and the query set are independently drawn from the same uniform distribution in $c$-dimensional unit hypercubes. The coordinates of the data points are never directly used and only the inter-object distances are taken as input to the algorithm. We set the indexed dataset size to 10,000, the validation set size to 100, and the query set size to 1,000. If not otherwise specified, the search range is set to retrieve about 5$\%$ objects from the database and he results are averaged over the 1,000 query objects.

### 3.1 Parameter Selection

There are three parameters in AdaIndex, i.e., the number of pivots in the level, $M$, the size of a separable set, $K$, and the number of levels in the structure, $H$. Because $H$ is



(a) (b) (c)

(d) (e) (f)

**Fig. 3.** Prune rules. (a) Rule 1. (b) Rule 2. (c) Rule 3. (d) Rule 4.

uniquely determined by $M$ and $K$, hereafter we only discuss how to tune $M$ and $K$. To find appropriate values of $M$ and $K$ with better search efficiency, we conduct a grid search process: Given a list of values for $M$ and a list for $K$, for each possible pair of parameters, we build an AdaIndex structure and test its performance on a validation set. The parameters with the best search performance are employed to build an index structure for evaluation on the test set.

In the following, we illustrate the above parameter tuning process of a 10-D dataset. The distance computations, disk accesses, and side computations against $M$ and $K$ averaged over the validation set are shown as contours in Fig. 3a-3c. In Fig. 3a, we can see that $M$ is the factor which most influences the number of distance computations at query time. The reason is obvious: $M$ determines the storage cost of the index structure, and intuitively, more stored information in the data structure will lead to more prevented distance computations. Fig. 3b shows the number of disk access for difference parameter settings. Here, we assume an ideal situation that a separable set can always be stored in a disk page. And thus it is obvious that the larger the value of $K$, the less disk accesses during the search. However, in a practical application where the object size is not ignorable to the size of a disk page, the contours may varies a lot. Consider the other extreme case that an object always takes one or more disk pages. Obviously, this time the number of disk accesses will correlate exactly to the number of distance computations. From the above analysis on the two extreme cases, we can see that for an application, whatever the ratio of the object size to the disk-page size, the actual contours of disk accesses will feature some combined properties in Fig. 3a and Fig. 3b. In fact, the larger the ratio of the object size to disk-page size, the flatter the slope of the contours. Fig. 3c presents the contour of side computations during the search. Similar as in Fig. 3a, the number of pivots is the deterministic factor for side computations. However, at this time, the effect of adding more pivots is minus – the more pivots employed the more side computations invoked during the search.

Above we have shown some basic principals for parameter tuning for a given evaluation criterion. It is worth noting that in a real application, computational cost of the search usually involves more than one factor from the above. However, the selection of parameters based on multiple criteria seems somewhat contradictive. Thus it will be interesting to explore some strategy on parameter tuning on multiple criteria. We adopt the weighting technique to solve this problem. For a given application, it is easy to estimate a pair of weight coefficients $\alpha$ and $\beta$ such that the overall computation cost of a query can be computed as

$$C = \alpha C_{\mathrm{dc}} + \beta C_{\mathrm{da}} + C_{\mathrm{sc}}. \tag{4}$$

Here, $C_{\mathrm{sc}}$ is the number of side computations which are taken as reference, $C_{\mathrm{dc}}$ the number of distance computations, and $C_{\mathrm{da}}$ the number of disk accesses. Base on the above definition, we can again make use of the grid search method to select a group of parameters which give optimal computational cost for the application. Fig. 3d shows the contours of the overall computational cost for $\alpha = 1,000$ and $\beta = 0$, i.e., the IO costs are neglected. We can see that for a quite wide range of parameters, i.e., $M \in [80, 500]$ and $K \in [40, 450]$, the search performance is near optimal. Fig. 3e shows the result when the IO costs are considered by setting $\beta = 10,000$. The acceptable range of parameter shrinks to be around $M \in [110, 370]$ and $K \in [250, 450]$.

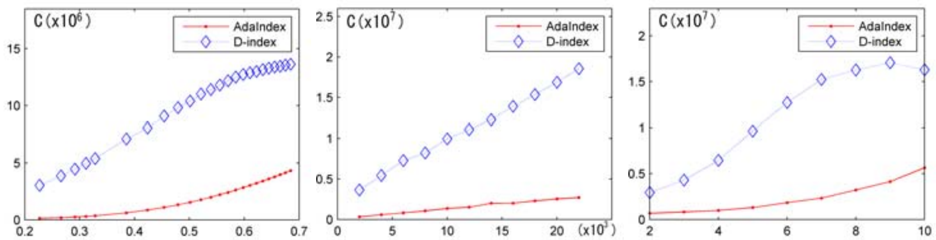
### 3.2 Performance Evaluation

In this section, we evaluate the search performance of AdaIndex on uniformly distributed data in $c$-dimensional Euclidean spaces. Fig. 3f shows the selected parameters for $c = 2, \cdots, 10$. We set $\alpha = 1,000$ and $\beta = 10,000$ all dataset. And for reference, we also report the results of D-index.

In the first experiment, we show how the search range $r$ influences the search performance of AdaIndex. The evaluation is done on a 5-D dataset. The search range is gradually increased with the answer set growing from about $0.2\%$ to $20\%$ of the indexed set. As shown in Fig. 4a, the overall cost of AdaIndex increases almost linearly with the search range, especially when the search range is small. For all the search ranges, AdaIndex results in much less overall cost than D-index since it avoids more distance computations during the search.

Then we test the performance of AdaIndex for datasets with varying sample size. AdaIndex and D-index are built on 5-D datasets up to size 20,000. In the evaluation, we fix the search range for retrieving about $5\%$ of the data. The curves of the overall cost against the dataset size are shown in Fig. 4b. It is obvious that the number of distance calculations for AdaIndex increases linearly as indexed data in the structure increases. Compared with D-index, the search performance of AdaIndex is more preferable.

The aim of the last experiment is to show the performance of AdaIndex on datasets with different dimensions. The algorithms are tested against datasets with dimensions up to 10. It is easy to learn from Fig. 4c that the search difficulty increases gradually



(a) (b) (c)

**Fig. 4.** Search performance measured by overall costs. (a) On a 5-D dataset with dynamic search range. (b) On a 5-D datasets with varying sample size. (c) On multi-dimensional datasets.

as the dimension grows because of the so called curse of dimensionality. In spite of the increased difficulty, AdaIndex again shows a much better performance than D-index.

Above all, AdaIndex has shown stable performance in a variety of simulated experiments. For all the tested datasets, it always has a perforable performance than D-index.

## 4 Conclusion

In this paper, we have presented the AdaIndex, a novel index algorithm for efficient similarity search in generic metric spaces. AdaIndex has incorporated the following principles to speed up the search. First, it borrows the idea from D-index to create a hierarchical data structure so that the search can be efficiently conducted in parallel mode. On an individual level, the objects are partitioned into compact clusters and are managed by the bucket structure for effective IO management. With the maximized search separable property among the buckets and the advanced prune rules, AdaIndex supports significant reduction in distance computations. Another very important property of the AdaIndex algorithm is its adaptivity for varying applications. Given a group of cost coefficients for different types of computational cost, we can easily apply the introduced grid search method for an index structure with optimal overall performance.

## References

1. Samet, H.: Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann Publishers Inc., San Francisco (2005)
2. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search – The Metric Space Approach. Series: Advances in Database Systems. Springer, Heidelberg (2006)
3. Dohnal, V., Gennaro, C., Savino, P., Zezula, P.: D-Index: Distance searching index for metric sata sets. Multimedia Tools and Applications 21(1), 9–33 (2003)
4. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. Theoretical Computer Science 38, 293–306 (1985)
5. Fukunaga, L., Narendra, P.M.: A branch and bound algorithm for computing k-nearest neighbors. IEEE Transactions on Computers 24(7), 750–753 (1975)
6. Kamgar-Parsi, B., Kanal, L.N.: An improved branch and bound algorithm for computing k-nearest neighbors. Pattern Recognition Letters 3(1), 7–12 (1985)

# The Application of Wavelet Neural Network Optimized by Particle Swarm in Localization of Acoustic Emission Source

Aidong Deng[1], Li Zhao[2], and Xin Wei[2]

[1] School of Energy & environment, Southeast University, Jiangsu Nanjing 210096, China
addseu@163.com
[2] School of Information Science and Engineering, Southeast University,
Jiangsu NanJing 210096, China
zhaoli@seu.edu.cn, nuptwx@163.com

**Abstract.** When using acoustic emission(AE) to locate the rub-impact source of rotating machinery, it is difficult to achieve exact source location for the effects of strong noise and waveform distortion. A neural network algorithm was presented to locate the AE source. In general BP wavelet neural network(WNN), it is a local search algorithm which falls into local minimum easily, so the probability of successful search is low. As an improved way, the particle swarm optimizer (PSO) algorithm was proposed to train the parameters of the WNN, then WNN based on PSO was used to locate the AE source. The localization experiment data of rub-impact AE signals was sampled from rotating test stand. The results show that the PSO algorithm is effective and the localization is accurate with proper structure of the network and the input parameters.

**Keywords:** particle swarm optimizer, wavelet neural network, acoustic emission.

## 1 Introduction

The Time Difference of Arrival (TDOA) location method is a usual approach to locate the position of a fault in the application of the acoustic emission technology in the fault location. The TDOA location method detects the time differences of arrival at different sensors from homologous acoustic emission signals, and then calculates the source location in accordance with the relationship of the space array between the sensors.

Use measured signals above the default threshold of the acquisition system to calculate the arrival time. The time is not only accordance with the parameters of acoustic emission instrumentations, the position of transducers, frequency dispersion, attenuation, noise interference and other factors on the acoustic emission signal when it is propagating, but also related by the experience of the engineering staff. In order to minimize the impact of man-made factors, it is worthy of designing a smart algorithm for source locating [1][2][3].

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 738–745, 2009.

In the smart algorithms, the Neural Network is a useful method. The Neural Network has its feature of self-organizing, self-adaptive, self-learning, and better robustness. With the rational structure of the network, right input samples, and enough training samples, the method can get the precise activity of the acoustic emission [4][5][6].

This paper brings in the Wavelet Neural Network module, uses the Particle Swarm Optimizer algorithm instead of the traditional decreasing gradient algorithm, optimizes the parameters of network, and implements the acoustic emission source location method. The experimental results show that the accuracy and efficiency are much higher than the traditional positioning methods.

## 2 Particle Swarm Optimizer Algorithm

The Particle Swarm Optimizer algorithm [7][8][9], which is based on the Swarm intelligence methods, originates from the group behavior of fish and birds. The solution of optimize problems looks as one point in the d-dimension, the point is designated a particle. One particle flies at a specific speed and adjusts the speed dynamically in accordance with the experiences of its own and peers. Every particle has its own goal function to decide the adaptive value, records particle best of its own and global best of all particles. As the iterations increases, the particle can converge gradually to a small extent, in accordance with the individual particle best and the groups' global best, get the optimal solution from final position when the iteration ends.

The algorithm of the elementary particle swarm optimization is as follows:

In the search space of d-dimension, m particles make up a group. The position of the i-th particle is $X_i = (x_{i1}, x_{i2}, \cdots, x_{id})$ ; the speed of the particle is a vector $V_i = (v_{i1}, v_{i2}, \cdots, v_{id})$ , the optimize position of the i-th particle is $\text{gbest} = (g_1, g_2, \cdots, g_d)$ , we can update the position of a particle by the formula (1) and (2):

$$v_{id}^{k+1} = \omega v_{id}^{k} + c_1 r_1 (\text{pbest}_{\text{i}}^{\text{k}} - x_{id}^{k}) + c_2 r_2 (\text{gbest}^{\text{k}} - x_{id}^{k}) \tag{1}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \tag{2}$$

In the above the formulas, k is the iterations, w is the inertia weight, which makes the particles maintain the inertia and have the capacity to search new areas. $c_1$, $c_2$ are the accelerating factors and non-negative constants. The flight weighting factors enable the particles fly to the optimal location $\text{pbest}_{\text{i}}$ and $\text{gbest}$. If they are small particles fly slowly to the target area, if large, the particles fly over the object location quickly; $r_1$ and $r_2$ are random numbers following [0,1] normal distribution, $V_i$ is the speed of flight and is limited by a maximum speed $V_{\max}$ ,If the speed at current dimension $v_{id}$ excesses $v_{\max d}$ ,which is the maximum speed at the dimension, $v_{id}$ is restricted to $v_{\max d}$ , $V_{\max}$ Decides the accuracy of searching in the solution space, if $V_{\max}$ is too large, particles will fly over the optimal solution. If $V_{\max}$ is too small, the particles are trapped into local optimal solution and can not process the global searching.

After a lot of experiments, we suggest the sum of $c_1$ and $c_2$ is around 4.0, set $c_1 \approx c_2 = 2.05$ . In general, $V_{\max}$ is equal to $X_{\max}$ ,as $V_{\max} = X_{\max}$ . $\omega$ ranges from 0.9 to 0.4. If $\omega$ decreases linearly as the process of iterative algorithm, the performance of algorithm will become more significant, $\omega$ decreases linearly as the formula (3):

$$\omega = \omega_{\max} - T\frac{\omega_{\max} - \omega_{\min}}{\mathrm{T}_{\max}} \tag{3}$$

$\omega_{\max}$ is the maximum value, $\omega_{\min}$ is the minimum value ; $T$ is the current iteration number and $\mathrm{T}_{\max}$ is maximum number of iterations, $\omega$ controls the impact of the previous speed on the current speed. If the $\omega$ is large, the impact of the previous speed is strong, so the ability of global search is strong. If the $\omega$ is small, the impact of the previous speed is small, so the ability of local search is strong. A good $\omega$ makes the solution not the local minimum.

## 3 Wavelet Neural Network

Wavelet neural network (WNN) [10][11], which is raised by Zhang in 1992. The main idea is using the wavelet function as a neuronal activation function and relating the Wavelet to the BP network. Since Wavelet transform has a good local time-frequency feature and multiresolution analysis capability, Wavelet neural network performs well at identification and approximates to any functions.

Wavelet transform are defined as follows:

$\psi(t) \in L^2(R)$ , ( $L^2(R)$ is the Square integrable space of real numbers, that is energy limited signal space), the Fourier transform of $\psi(t)$ is $\Psi(\omega)$ . If $\Psi(\omega)$ meets a condition as follows

$$C_\Psi = \int_R \frac{|\Psi(\omega)|}{|\omega|} d\omega < \infty \tag{4}$$

$\psi(t)$ is the Basic Wavelet or Mother Wavelet. After stretching and shifting $\psi(t)$ , we can get a Wavelet sequence

$$\psi_{a,b}(t) = |a|^{-1/2} \psi\left(\frac{t-b}{a}\right) \qquad a,b \in R; a \neq 0 \tag{5}$$

$a$ is the stretching factor, $b$ is the shift factor. If the function $f(t) \in L^2(R)$ , Wavelet transform of $f(t)$ is defined as followed:

$$W_f(a,b) = \langle f, \psi_{a,b} \rangle = |a|^{-1/2} \int_R f(t)\psi\left(\frac{t-b}{a}\right)dt \tag{6}$$

Inverse transform is defined as followed:

$$f(t)=\frac{1}{C_\psi}\int_R \frac{1}{a^2}W_f(a,b)\psi\left(\frac{t-b}{a}\right)dadb \tag{7}$$

The discretization of $f(t)$ is:

$$f(t)=\sum_{i=1}^{k}\omega_i\psi(\frac{t-b_i}{a_i}) \tag{8}$$

$k$ is the number of wavelets.

The diversity and complexity of the Wavelet function construction decide the diversity and complexity of the wavelet network structure. At present, the majority of scholars at home and abroad have adopted compact structures. In practice, we generally make the number of hidden layers one, which is a three layer of neural network with a single hidden layer. In the equation (8), a signal function $f(t)$ can be fitted by the wavelet in the form of linear superposition, the network structure as Fig.1.

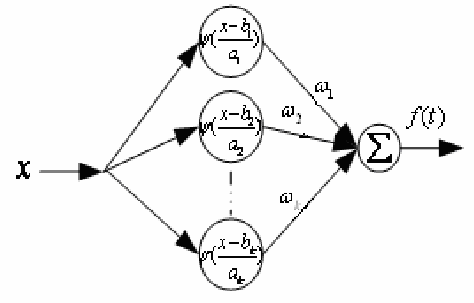

$\psi(\frac{x-b_1}{a_1})$
$\psi(\frac{x-b_2}{a_2})$
$\psi(\frac{x-b_k}{a_k})$
X
$\omega_1$
$\omega_2$
$\omega_k$
$\Sigma$
$f(t)$

**Fig. 1.** The structure of Wavelet network

$a_i$ 、 $b_i$ is unsetted, $\omega_i$ is the weight between the hidden layer and output layer, $\psi((t-b_i)/a_i)$ is the output value of the input node. The parameters need to be determined are the number of hidden nodes $k$, scale factor $a_i$, shift factor $b_i$ and the weight between the hidden layer and output layer $\omega_i$.

## 4 The Application of Particle Swarm Optimization Algorithm in Wavelet Neural Network

In the BP wavelet neural network, the most common learning algorithm is BP algorithm, which is based on gradient information to adjust the connect weights; it may fall into local extreme points easily. Particle swarm optimization algorithm has a characteristic of fast convergence and high robustness, global search capability. If use it to optimize the neural network connection weights, we can overcome the local extreme points of BP neural network. In addition, the algorithm improves the convergence rate of neural networks, with a pan-neural network capability and learning ability.

When using particle swarm optimization to train neural network, we define of the vector position of particle swarm at the first. As shown in Figure 1 of the wavelet neural network, the number of particle swarm initialize to 30, the vector position of each particle is:

$$x(i)=[\omega_{i1},\cdots\omega_{ij},a_{i1},\cdots a_{ij},b_{i1},\cdots b_{ij}]\quad i=1,2,\cdots 30 \tag{9}$$

$j$ is the number of hidden layer neurons. Fitness function for the neural network is the mean square error indicator, and the formula is as follows:

$$J(k,i)=\sum_{m=1}^{n}(y_{m,i}-\hat{y}_{m,i}^{k})^2,k=1,2,\cdots,N \tag{10}$$

$J(k,i)$ is the $i$ particle fitness value after $k$ times iterated, $n$ is the number of training samples, $y_{m,i}$ is the ideal output value for the network after the $m$ sample input of the $i$ particle, $\hat{y}_{m,i}^{k}$ is the practical output value for the network after the $m$ sample input of the $i$ particle, $k$ is the iterate number, $N$ is the max number of iterate.

## 5 Experiment Analysis

The test bed of Rotor system acoustic emission friction is as shown in Fig. 2. The input voltage of the motor regulates the rotational speed, the semi-flexible shaft connects the electric motor and the shaft section, the sliding bearing chock supports the rotor. A Mobile friction device is installed at the base of the Test-bed. the Mobile friction device locates at the space between shaft blocks 1 and 2, a retractable bolt Installs on the side of the screw, along the center of radial axis, the acoustic emission signals will be excited from the friction between the rotors by adjusting the bolts.

The Two acoustic emission sensors installs separately on the shaft blocks 1 and 2, space coordinates are (x1, y1)= (0, 0) and (x2, y2) = (50, 0).The acoustic emission is on the shaft, whose distance from the sensor is 1=20cm. we set the sampling frequency 1MHz, the number of points is 16384. Fig. 3 is acoustic emission signals received by the two sensors when the rotors are rubbing.

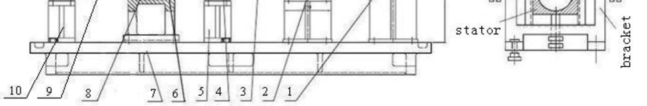


**Fig. 2.** Test bed of the Rotor friction Acoustic Emission. 1- electromotor ; 2-accelerate box ; 3- shaft joint ; 4- bearing ; 5- bearing base 1 ; 6- rub-impact set ; 7- base ; 8- turntable ; 9- axle ; 10- bearing base 2.

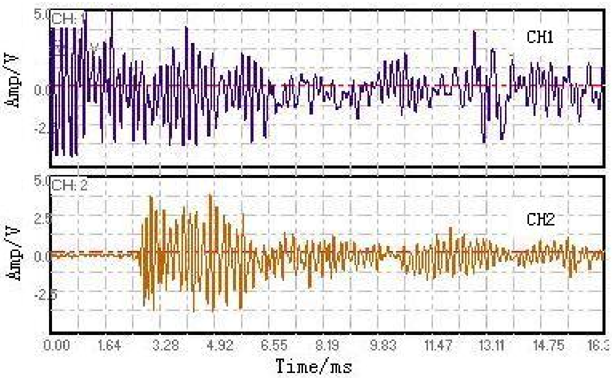


**Fig. 3.** Two-way acoustic emission signals

We get Two-way signals' amplitude and energy; use the ratio of the coordinates of the sensor and the received signal energy as the input of the neural network. In the experiment, we set the rotating shaft as the *x* axis; the perpendicular to the direction of the rotating shaft is the *y* axis. The direction of acoustic emission signals is the *x* axis, and the neuron output number of the network is 1.

We rub the rotors twenty times and use the date as the input of the network to train the neuronic network. From the Fig. 4, if the Hidden layer neuron number is less than 10, the network is not stable and the correct rate of the output is volatile. As the Number of hidden layer neurons increases, both the network and the correct rate of the output will be stable. We set the number of hidden layer neurons 10 for the accuracy and efficiency of the network.

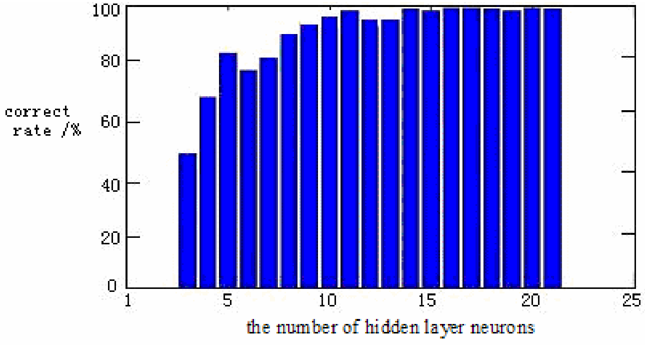


**Fig. 4.** The relationship between the correct rate and the hidden layer

We use the network above to experiment on 5 different locations of rotor friction. The results give us the information about the Performance of Particle Swarm Optimization and Wavelet neural network. Network prediction results and actual results are listed in the Table 1.

**Table 1.** Comparison betwwen Network prediction results and actual location

| Actual location (away from the sensor 1)/cm | Network prediction results/ cm | Error rate/ % |
|---|---|---|
| 5 | 5.09 | 1.80 |
| 10 | 9.78 | 2.20 |
| 20 | 19.83 | 0.85 |
| 30 | 28.61 | 4.60 |
| 40 | 40.29 | 0.73 |

Wavelet neural network with the Particle Swarm Optimized makes the correct rate high. The error rate is below 4%. The error rates in table 1 are irregular, probably because the training samples are not enough. Some parameters of the network not the best brings about the high error rates. The lack of de-noising makes the output error inevitable.

## 6 Conclusions

In this paper, we take the particle swarm algorithm in place of the traditional gradient descent method, which optimizes the parameters of the wavelet neural network. We apply the new algorithm in the acoustic emission source location experiment. we use the characteristic parameters of acoustic emission, set the ratio of the sensor coordinates to signal energy as a neural network input, the error variance when increase the number of hidden layer neurons helps determine the optimal number of hidden layer neurons.

The experimental results show that the network designed above is not only good at the prediction of the acoustic emission source location, but also has a satisfied efficiency. It is a new approach for acoustic emission source location. Particle swarm optimization algorithm has the local minimum problem, further research is needed to reduce the probability of local minimum value.

## Acknowledgement

This paper is supported by the National High Technology Research and Development of China (863Program)(2007AA04Z434).

## References

1. Gorman, M.R.: AE source orientation by plate wave analysis. Journal of Acoustic Emission 10(4), 53–58 (2000)
2. Surgeon, M., Wevers, M.: Modal analysis of acoustic emission signals from CFRP Laminates. NDT and E International 32(6), 311–322 (1999)
3. Morscher, G.N.: Modal acoustic emission of damage accumulation in a woven SIC/SIC composite. Composites Science and Technology 59(6), 687–697 (1999)

4. Bodine, H.L., Dawes, R.L., Henley, S., et al.: Detecting crack growth in metal structures using temporal processing and the parametric avalanche stochastic filter neural network. Intelligent Engineering Systems.Through Artificial Neural Networks 5(11), 467–472 (1995)
5. Hironobu, Y., Kyoji, H.: Analysis of artificial acoustic emission waveforms using a neural network. Acoustic Emission 10(5), 35–41 (1995)
6. Tian, Y., Lewin, P.L., Davies, A.E.: PD pattern identification using acoustic emission measurement and neural networks. IEEE Conference Publication (5), 541–543 (1999)
7. Shi, Y., Eberhart, R.C.: A modified swarm optimizer. In: IEEE International Conference on Evolutionary Computation, Anchorage, AK USA, pp. 69–73. IEEE, Los Alamitos (1998)
8. Eberhart, R.C., Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Congress on Evolutionary Computation, CA USA, La Jolla, pp. 84–88 (2000)
9. Geng-huang, Y., Bo-ying, W.: Identification of Power Quality Disturbance Based on QPSO-ANN. In: Proceedings of the CSEE, vol. 28(10), pp. 123–129 (2008) (in Chinese)
10. Qinghua, Z., Albert, B.: Wavelet Networks. IEEE Trans. on Neural Network 3(6), 889–898 (1992)
11. Hai-long, B., Guang-ju, C., Tian-jun, D.: A Method of Time-varying Harmonic Detection Based on the Wavelet Neural Network. In: Proceedings of the CSEE, vol. 28(7), pp. 104–109 (2008) (in Chinese)

# Speaker Recognition Based on GMM with an Embedded TDNN

Cunbao Chen and Li Zhao

School of information science and engineering
Southeast University, Nanjing, Jiangsu, 210096, China
chencunbao@gmail.com

**Abstract.** A modified GMM with an embedded TDNN is proposed to speaker recognition. The model integrates the merits of GMM and TDNN. TDNN is used to digest the time information of the feature sequences, and through the transformation of the feature vectors the model makes the hypothesis of variable independence which maximum likelihood needed more reasonable. In the process of training, GMM and TDNN are trained as a whole and the parameters of GMM and TDNN are updated alternately. Experiments show that the proposed model improves accuracy rate against baseline GMM at all SNR with a maximum to 22%.

**Keywords:** Speaker Identification, Gaussian Mixed Model, Time Delay Neural Network, Embedded.

## 1 Introduction

In a large number of speaker recognition methods [1], GMM has been paid more and more attention [2][3], it has become the mainstream method of Speaker recognition with its advantage of high recognition rate, easy training. SVM and factor analysis based on GMM super-vector [4][5][6] represent the latest achievements of GMM methods. Because of good ability for data distribution, GMM can approach any distribution model as long as there are enough Gaussian items and training data. However, some problems exist when using GMM in practice. First, GMM has never taken advantage of time information of speaker speech and the results of training and recognition are unrelated to order of input vectors. Secondly, it is obviously unreasonable on the assumption that feature vectors are mutually independent when GMM are trained. It should be said that neural networks in speaker recognition has also held an important position, Multi-Layer Perceptron (MLP), Radial Basis Function (RBF) [7] and auto-associative neural network (AANN) [8] have been successfully applied to speaker recognition, especially the time-delay networks (TDNN) is widely used in signal processing, speech recognition and speaker recognition [9][10][11] and it takes full advantage of time information of the feature sequences. Neural network learns and transforms the vectors, so that transformed vectors can approach target vectors in some way, and the approximation criteria is usually minimum least square method.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 746–753, 2009.

This paper takes full account of GMM and TDNN with their respective advantages, put forward a combined method of GMM and TDNN to identify speaker, that is to embed the TDNN into GMM and uses maximum likelihood probability as the common criteria for training GMM and TDNN, thus TDNN learns the time information of vectors and transmits it to GMM. Also the influences of assumption where vectors are independent can be reduced.

The rest of this paper is organized as follows. The GMM and TDNN in Speaker Identification are introduced in Section 2. Section 3 proposes the model of GMM with embedded TDNN, together with a training algorithm. In the next Section, experimental data, results and corresponding discussions are given. Finally, Section 5 gives a summary and conclusions.

## 2 GMM and TDNN in Speaker Recognition

GMM can be seen as a HMM of one state. The probability density function of a M-order GMM is consist of weighted summation of M Gaussian probability density functions, which can be expressed as [2][3]:

$$p(x_t|\lambda) = \sum_{i=1}^{M} p_i b_i(x_t) \tag{1}$$

The $x_t$ here is a D-dimensional random vector; $b_i(x_t)$ , $i = 1,2,...,M$ are members of density; $p_i , i = 1,2,...,M$ are the mixing weights. Each of the members of density is the Gaussian function of both $u_i$ as a mean vector and $\sum_i$ as a covariance matrix, which can be expressed as:

$$b_i(x_t) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(x_t - u_i)^T \Sigma_i^{-1}(x_t - u_i)\right\} \tag{2}$$

Where $p_i$ must satisfy the condition: $p_i > 0, \sum_{i=1}^{M} p_i = 1$

Complete Gaussian mixture density is consist of the parameterized mean vectors, covariance matrixes and mixing weights of all members of density. It can be defined as follows:

$$\lambda = \{(p_i, u_i, \Sigma_i), i = 1,2,..., M\} \tag{3}$$

Considering the small amount of data in the general training and identification, so assumption of diagonal covariance matrix of every Gaussian mixture density is reasonable, and EM is usually used to train parameters of GMM.

Time Delay Neural Network (TDNN) has been widely used in speech recognition and speaker recognition, it has the ability to compare and associate the current input and previous input. Therefore, network input must be in accordance with the order of time sequence. There are TDNN networks with feedback and without feedback, and this paper mainly uses the network without feedback. Delayed vectors are transformed in non-linear manner, and the transform results are linearly weighted as output. TDNN is a Multi-layer Perceptron network and therefore the method of training Multi-layer Perceptron network also applies to training TDNN network.

## 3 GMM with the Embedded TDNN

The proposal of embedding TDNN into GMM is advanced based on the merits of GMM and TDNN. It takes advantage of GMM's ability to express data distribution and learning ability of TDNN to data structure. TDNN learns time information of vector sequences and balances the requirement of variable independent which is needed by maximum likelihood probability. They are trained as a whole, commonly use maximum likelihood probability. In order to explain the method detailed, the following aspects in training and recognition model, training method and convergence proof of TDNN will be deduced and illustrated.

- **Training and Identification Model**

Fig.1 shows the training and identification model with embedded TDNN for speaker identification, which is different from baseline GMM model in the training and recognition.

In training, delayed vectors of target speaker is sent to the TDNN, the TDNN learns the structure of vector set and then extract time information of vector sequence. After that, TDNN provides the learning results to GMM by residual error (i.e., the error of input of vector and the output of the TDNN), EM is used to train GMM, and then using the BP algorithm to update the weight coefficients of TDNN. Training and learning guidelines of TDNN and GMM model are maximum likelihood probability.

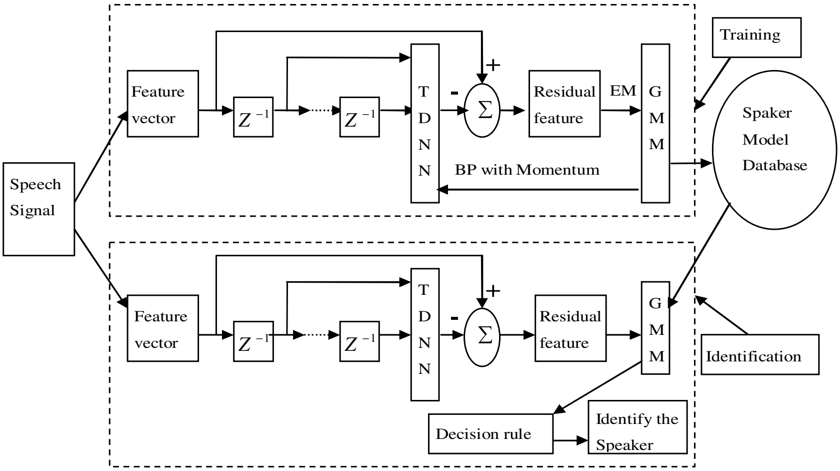


**Fig. 1.** The training and identification model of GMM with embedded TDNN

By this way, residual distribution is likely to go toward enhancing the likelihood probability of target speaker.

In identification, after delayed, vectors come into TDNN network for identifying. Because the TDNN has learned the structure and time information of feature vector space, the TDNN network would carry out appropriate transform and then made available to the GMM model when inputting vector. TDNN transform can enhance the likelihood probability of target model, simultaneously, can reduce the likelihood probability of non-target model.

- **Training Methods**

A two-stage approach is used to train the model. The process of the TDNN training and GMM training are alternating. EM method is used when training GMM. Because TDNN network is a kind of multi-layer Perceptron network (MLP), BP method with momentum is used when training TDNN.

Training process described as follows:

1. Determine the structure of GMM model and TDNN network;
2. Give convergence conditions and the largest number of iterations;
3. Select the initial parameters of TDNN network and GMM model randomly;
4. Fix the TDNN parameters, input feature vectors and get all of the residual vectors of the TDNN;
5. Use EM method to modify the weights, means and variances of Gaussian distributions;
6. Fix the GMM parameters, an expression of likelihood probability will be got, then the TDNN parameters can be modified by means of BP method with momentum;
7. If the training conditions or number of iterations are satisfied, stop, or turn to 4.

- **Derivation of the Formula of Modifying TDNN Neural Network Parameters**

Because the two-step iterative method for model parameters is adopted, in the iteration of neural network parameters, the weights, the mean vectors and variances of the Gaussian distributions are fixed. The negative logarithm is used and the result is expressed as follows:

$$\omega_{ij}^{*} = \arg\min_{\omega_{ij}} \left(-\sum_{t=1}^{N} \ln p((x_t - o_t) \mid \lambda)\right) \tag{4}$$

Here $p((x_t - o_t) \mid \lambda)$ is explained in (1) and $o_t$ is the neural network output. The BP method with momentum [12] (shown as follows) can speed up the convergence process again and again, and it can deal with local minimum problem better also.

$$\Delta\omega_{ij}^{k}(m+1) = \gamma\Delta\omega_{ij}^{k}(m) - (1-\gamma)\alpha \frac{\partial F(x)}{\partial \omega_{ij}^{k}} \Big|_{\omega_{ij}^{k} = \omega_{ij}^{k}(m)} \tag{5}$$

Here $\Delta\omega_{ij}^{k}(m+1) = \omega_{ij}^{k}(m+1) - \omega_{ij}^{k}(m)$, $\gamma$ is the momentum coefficient, $\alpha$ is the iterative step size, whose value can be selected according to the actual situation, $m$ is the iteration number and k represents the layer number of the neural network. As

$F(x) = -\ln p((x-o)|\lambda)$ (here a single sample instance is used) shows, the subscript t is omitted for convenience.

Calculate $\frac{\partial F(x)}{\partial \omega_{ij}^{k}}$ as follow:

$$\frac{\partial F(x)}{\partial \omega_{ij}^{k}} = \frac{\partial F(x)}{\partial y_i^{k}} \frac{\partial y_i^{k}}{\partial \omega_{ij}^{k}} \tag{6}$$

$$\frac{\partial y_i^{k}}{\partial \omega_{ij}^{k}} = o_j^{k-1} \tag{7}$$

The output layer $\frac{\partial F(x)}{\partial y_i^{k}}$ :

$$\begin{aligned} \frac{\partial F(x)}{\partial y_i^{k}} &= -\frac{1}{p((x-o)|\lambda)} \frac{\partial p((x-o)|\lambda)}{\partial o_i^{k}} \frac{\partial o_i^{k}}{y_i^{k}} \\ &= -\frac{f'(y_i^{k})}{p((x-o)|\lambda)} \sum_{n=1}^{M} p_n c_n \left(\frac{a_n(x-o-u_n)}{\sigma_{n,i}^{2}}(x_i - o_i - u_{n,i})\right) \end{aligned} \tag{8}$$

Here:

$$\begin{aligned} a_n(x-o-u_n) &= \exp\left(-\frac{1}{2}(x-o-u_n)^T \Sigma_n^{-1}(x-o-u_n)\right) \\ c_n &= \frac{1}{(2\pi)^{D/2}|\Sigma_n|^{1/2}} \end{aligned} \tag{9}$$

The hidden layer $\frac{\partial F(x)}{\partial y_i^{k}}$ :

$$\frac{\partial F(x)}{\partial y_i^{k}} = \sum_j \frac{\partial F(x)}{\partial y_j^{k+1}} \frac{\partial y_j^{k+1}}{\partial y_i^{k}} = f'(y_i^{k}) \sum_j \frac{\partial F(x)}{\partial y_i^{k+1}} \omega_{ji}^{k+1} \tag{10}$$

Due to back inversion, the output layer $\frac{\partial F(x)}{\partial y_i^{k+1}}$ is known. Put it into (10), $\frac{\partial F(x)}{\partial y_i^{k}}$ .will be calculated.

Finally, simple proof that the training method proposed in the paper will has a local maximum point of the model is given. As we know, the likelihood probability is monotone non-drop in each EM training progress. If an appropriate step size and momentum coefficient are selected to make sure that the likelihood probability after iteration is monotone and non-drop in TDNN, by an Advanced Mathematics theorem, Monotone non-descending sequence must be limitary, we can prove that the training process does converge to a local maximum point.

Certainly, in order to avoid premature convergence to local maxima, we can select a bit bigger step size at the beginning, in which there has advantage if there is oscillation in the process of training, but we must put the iterative step size under control late, otherwise the training process will be too long or even divergence.

Regardless of GMM or TDNN, the above methods are theoretically obtained point of local maxima. Therefore, it is necessary for us to train from more initial values and step sizes so that better model parameters can achieve.

## 4 Experiment

Speech used in the experiment comes from 1conv4w-1conv4w set of NIST SRE 2006, 167 target speakers are selected, in which there are 73 men and 94 women. About five minutes speech for each person are chosen as training speech, 8-12 seconds speech as testing speech and closed-set tests are used, so there are about 21000 tests.

In order to test the improved results in the noisy environment, the noise data is selected from Japan Electronics Association standard noise database of moving cars (2000cc Group, roads). This noise must be in accordance with the signal to noise ratio (SNR) superposition into 1conv4w-1conv4w speech, and then generate speech with the noise.

The method based on energy and zero-crossing rate is used to carry out silent detection, and spectral subtraction is used in the process of de-noising. After above, a pre-emphasis through the filter of $f(Z) = 1 - 0.97Z^{-1}$ is utilized to speech data, then speech data is divided into separate frames of 20ms length and 10ms window shift. Adding the Hamming window to the framed signal, 19-dimensional MFCCs are formed.

In the experiments, the TDNN that dimension ratio of hidden layer neurons and input layer is 3:2 is utilized, with non-linear activation function using S function. The neural network of inertia coefficients is 0.8 when training, and GMM adopts diagonal covariance matrix and the least value of $\sigma^2$ is 0.001. Because the method does not affect the follow-up treatment, so we just compared with the baseline GMM.

Speaker identification, different from the speaker Verification, judges who is the target speaker according to the provision of speech, that is to say, speaker identification generally adopts correct recognition rate or error recognition rate as the evaluation criteria of identify effect [2], $R = N_v / N_t$,where $R$ for the correct recognition rate, $N_v$ for correctly identifying frequency, $N_t$ for the total number of tests.

The experimental results shown in Table 1, Table 2, Table 3, Table 1 shows the improvement results to change the number of mixed items of GMM under the condition of 1conv4w-1conv4w in noise-free, M is the mixed number. Table 2 shows the improvement results to change the number of input vectors of the TDNN network, and N is number of input vectors. Table 3 shows the improvement results in different

SNR conditions. The number of input vectors in TDNN input layer is four in Table 1, Table 3, and the mixed number of GMM is eighty in Table 2, Table 3.

**Table 1.** Comparative data in 1conv4w-1conv4w without noise

| M<br>Recognition rate | 16 | 32 | 48 | 60 | 80 |
|---|---|---|---|---|---|
| **baseline GMM** | **84.7** | **90.4** | **93.3** | **95.3** | **96.3** |
| **TDNN-GMM** | **90.4** | **93.8** | **96.7** | **97.6** | **98.1** |

From the Table 1, we know that after embedded TDNN in GMM, Model identification effect has indeed improved and the effectiveness is more obvious when fewer numbers of GMM mixed items because learning of neural network becomes better when fewer sub-class categories exist.

**Table 2.** Comparative data without noise at various inputting numbers of vector

| N<br>Recognition rate | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **TDNN-GMM** | **90.2** | **94.3** | **96.1** | **98.1** | **95.2** | **92.1** |

From the Table 2, we can see that the difference of identification effect is somewhat bigger to different number of input features. To small number of input vectors, because the smoothing effect is not enough and affect the identification effect, the results of identification can not catch up the baseline system; but the identification of effects has the same with baseline system as the input number is three; the effect of identification has overtake the baseline system when the input number is four; however, the effect of identification declines when the number of input vectors becomes bigger, that's because the number of input vectors can lead into increasing number of parameters so that the model can not receive enough training and results in declining the identification effect.

**Table 3.** Comparative data within the moving car

| SNR<br>Recognition rate | 0 | 5 | 10 | 15 | 20 | 25 | ∞ |
|---|---|---|---|---|---|---|---|
| **baseline GMM** | **30.2** | **44.8** | **63.1** | **78.5** | **88.5** | **93.4** | **96.3** |
| **TDNN-GMM** | **35.1** | **53.6** | **76.9** | **88.2** | **93.1** | **96.7** | **98.1** |

From the Table 3, we can see the improvement effect of the proposed method is not very good when the noise is so power, that's maybe because of powerful noise affect learning results of neural network, but when the signal to noise ratio at a certain range, the improvement effect is good.

## 5 Conclusion

In this paper, the model embedding a TDNN into GMM is proposed, it is equivalent to transform the characteristics domain and model domain of baseline GMM simultaneously, which can lead into learning vector's time information and increasing the likelihood probability of target speaker, and balances the requirements of variable independence that maximum likelihood probability need. In the noise environment, it can better eliminate the effect of noise. A two-stage iterative method is utilized to better solve the model of learning method. When the weight coefficients of neural network's output layer are 0, the model degraded to baseline GMM. The experimental results show that speaker identification effects of the method, in without noise environment and noise environment, become better than the baseline GMM.

## References

1. Bimbot, F., Bonastre, J.F., et al.: A Tutorial on Text-Independent Speaker Verification. EURASIP Journal on Applied Signal Processing 4, 430–451 (2004)
2. Reynolds, D.A., Rose, R.C.: Robust Text-independent Speaker Identification Using Gaussian Mixture Speaker Models. IEEE Trans. Speech Audio Processing 3, 72–83 (1995)
3. Reynolds, D., Quatieri, T., Dunn, R.: Speaker Verification Using Adapted Gaussian Mixture Models. Digital Signal Processing 10, 19–41 (2000)
4. Campbell, S.D.E., Reynolds, D.A.: SVM based speaker verification using A GMM supervector kernel and NAP variability compensation. In: Proceedings of ICASSP, pp. 97–100 (2006)
5. Yin, S.-C., Rose, R., Kenny, P.: A joint factor analysis approach to progressive model adaptation in Text-Independent speaker verification. IEEE Trans. on Audio, Speech and Language Processing 15, 1999–2110 (2007)
6. Dehak, N., Dehak, R., Kenny, P., Dumouchel, P.: Comparison between Factor Analysis and GMM Support Vector Machines for Speaker Verification. In: Proceedings of IEEE Odyssey 2008 - The Speaker and Language Recognition Workshop (Odyssey 2008), Stellenbosch, South Africa, January 21-25 (2008)
7. Mak, M.W., Allen, W.G., Sexton, G.G.: Speaker identification using multilayer perceptron and radial basis function networks. Neurocomputing 6(1), 99–117 (1994)
8. Murty, K.S.R., Yegnanarayana, B.: Combining evidence from residual phase and MFCC features for speaker recognition. IEEE signal processing letters 13, 52–55 (2006)
9. Bennani, Y., Gallinari, P.: On the use of TDNN-Extracted Features Information In Talker Identification. In: Proceedings of ICASSP, pp. 385–388 (1991)
10. Dev, A.: Effect of retroflex sounds on the recognition of Hindi voiced and unvoiced stops. AI & SOCIETY 23(4), 603–612 (2009)
11. Zhou, H., Taj, M., Cavallaro, A.: Target Detection and Tracking With Heterogeneous Sensors. IEEE Journal of Selected Topics in Signal Processing 2, 503–513 (2008)
12. Vogl, T.P., Mangis, J.K., Aigler, A.K.: Accelerating the convergence of the backprogagating method. Biological Cybernetics 59, 256–264 (1988)

# Finding Appropriate Turning Point for Text Sentiment Polarity

Haipeng Wang, Lin Shang, Xinyu Dai, and Cunyan Yin

State Key Laboratory for Novel Software Technology, Nanjing University
Department of Computer Science and Technology, Nanjing University
wanghaipeng@ai.nju.edu.cn, {shanglin,mtlab,yincy}@nju.edu.cn

**Abstract.** Sentiment analysis has attracted more and more attention in recent years. Neural network can be trained to calculate the sentiment orientation value of the text. After getting the value, a turning point is used to identify the text polarity. The midpoint 0.5 is often used as the turning point, however, we show in this paper that the midpoint is not always good for getting the highest classification precision. In the paper, three methods are proposed to find the appropriate turning point. We prepare the book review from Amazon.com and experiment our three methods. Neural Network classifier is employed in experiments and the results show the better precision compared with midpoint method.

**Keywords:** sentiment polarity, neural network, turning point.

## 1 Introduction

Sentiment analysis has attracted a great deal of attention for its potential applications. For instance, the interest that individual users show in online opinions about products and services, and the potential influence such opinions wield, is something to which vendors of these items are paying more and more attention [1].

Contributions for sentiment analysis focused mostly on interpretation of metaphor, narrative, point of view, affect, evidence in text, and related areas [1]. And text sentiment classification has been one interesting branch. Most prior work on the specific problem of categorizing expressly opinionated text has focused on the methods to get the candidate features from the text corpus [2–4], select features[5, 6] and train different classifiers [7–9].

In text sentiment classification, we try to find a function, such as

$$f(i_1, i_2, \cdots, i_n) = y \tag{1}$$

to calculate the sentiment orientation value of this text, here $i_1$, $i_2$, $\cdots$, $i_n$ are the values of text features, $i_j \in \{0, 1\}$, $j = 1, 2, \cdots, n$, if the feature $j$ found in the text $i$, then $i_j = 1$, otherwise, $i_j$=0. $y$ is the sentiment orientation value of the text, $y \in [0, 1]$. There may be a question about how to get this function used to calculate the sentiment orientation. Neural networks have been used for text

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 754–761, 2009.

categorization [11–13]. And, there has been some work using neural network to handle the sentiment [14, 15]. In these methods, neural network is employed to fit the function of $f$. After getting the value of $y$, the polarity of the text can be identified. An appropriate turning point $x$ should be decided to determine whether the polarity of the text is positive or negative, if $y \geq x$ then the text is labeled as a positive evaluation, otherwise, it is labeled as a negative one. In general, the midpoint 0.5 is set as the value of $x$ . However, our experiment illustrate that 0.5 is not always the best choice to classify the sentiment text. In this paper, we present three methods to get the appropriate value of $x$ as the turning point. Based on the neural network, we apply these methods for determining the polarity of book review in Amazon.com. We show that the proposed methods provide good precision compared with midpoint method and Naive Bayesian classifier.

The rest of this paper is organized as follows: Section 2 presents three algorithms for getting the turning point. Section 3 describes the experiments of these algorithms and compares the results of these algorithms. Section 4 concludes the paper.

## 2 The Algorithms to Get the Turning Points

In sentiment polarity classifier which deals with binary classification, people use the default value 0.5 as the turning point. If the sentiment orientation value of the text is greater than or equal to 0.5, we label the positive sentiment to the text, otherwise, we label the text as a negative one. However, if we obtain the turning point in this default way, we would neglect important information in data itself. In this section, we propose three algorithms using the distribution of data to get the best turning point.

### 2.1 Adjust the Upper and Lower Bounds (ADJUST)

**Preliminaries.** This algorithm bases on the hypothesis that most positive text spread in the span near the upper bound 1, the farther away from 1, the fewer the number of positive text and the lower the density of the positive text is. In the negative text set, most negative texts spread in the span near the lower bound 0, the farther away from 0, the fewer the number of negative text, the lower the density of the text is.

**Description of the algorithm.** First, we can get the upper and lower bounds for positive and negative text polarity values in the training set. Second, we divide the range into the given sub-intervals and get the density of the values in every sub-interval. Third, the boundary value N is set, if the density is lower than N, we can conclude that it's sparse in the sub-interval. At last, we can adjust the upper and lower bounds for positive and negative text sets to make sure that in their span, the density is higher than N in every sub-interval. The algorithm is described in detail as follows.

*Adjust the upper and lower bounds*

```
Input: the vector of positive text in training set pos_vec, the
vector of negative text in training set neg_vec
Output: the sentiment vector of the text of test set Algorithm
description:
pos_min_max = minmax(pos_vec) %get the interval of positive text
neg_min_max = minmax(neg_vec) %get the interval of negative text
%split the interval into twenty sub-interval
pos_interval = (pos_min_max(2)-pos_min_max(1))/20
pre_length = 0;
for i = 1:20
  %calculate the density of the sub-interval
  length = length(pos_vec(find(pos_vec>(pos_min_max(2)
                                         -pos_interval*i))))
  if(length-pre_ length < N)
    pos_min_max(1)=pos_vec(600-length+1)
    break;
  else
    prelength = length;
%we can use the similar method to adjust the upper bounds of
%negative text set, we don't repeat here
mid_1 = (pos_min_max(1)+neg_min_max(2))/2;  %mid_1 is the
                                            %turning point
for i = 1:Num   %Num is the number of test set
    if a_1(i)>=mid_1      %a_1 is the vector of test set
          a_1(i)=1;
      else
          a_1(i)=0;
```

### 2.2 Find the Highest-Accuracy Point of Training Set (ACCURACY)

**Preliminaries.** This algorithm bases on the hypothesis that the point that makes the training set produce the highest accuracy should also get the highest precision for the test set, because the training set and the test set have the same distribution.

**Description of the algorithm.** First, we set $n$ points in [0,1]. Then, we calculate the precision for every point in the training set. At last, we set the point that has the highest precision in training set as the turning point $x$. The value of $x$ is calculated as $argmax(p(x_i)|i = 1, 2, \cdots, M)$ where $p(x_i)$ is the precision of training set when we use $x_i$ as the turning point. The algorithm is described in detail as follows.

*Find the highest-accuracy point in training set*

```
Input: the vector of positive text in training set pos_vec, the
vector of negative text in training set neg_vec
Output: the sentiment vector of the text of test set Algorithm
description:
interval = 1/n;%split the interval [0.05,0.95] into n sub-interval
pos_dis = 0;  neg_dis = 600;
for i=0:n
  pos_dis = [pos_dis,length(pos_vec(find(pos_vec>(interval*i))))]
  neg_dis = [neg_dis,length(neg_vec(find(neg_vec<(interval*i))))]
end
dis = pos_dis+neg_dis
[value,position] = max(dis) mid_2 = (position-1)*interval;
for i = 1:Num   %Num is the number of test set
    if a_1(i)>=mid_2      %a_1 is the vector of test set
          a_1(i)=1;
    else
          a_1(i)=0;
```

### 2.3 Similar *K*-Nearest Neighbor (NEIGHBOR)

**Preliminaries.** The idea of this algorithm is similar to the $k$-nearest neighbor algorithm. The dissimilarities are dwelled on as follows: First, the distance measurement is the difference between the polarity values of the texts. Second, parameter $k$ is the distance away from the test text, rather than the number of the nearest in the test text.

**Description of the algorithm.** First, we calculate the training set, and set the parameter of $k$. Then we design the value of $x$, we calculate the number of positive text and negative text in training set in the interval $[x - k, x + k]$.At last, we confirm the polarity of the text. In the algorithm, we use the rule that in the interval, if the number of positive text is more than that of negative text, the polarity of the test text spread in the interval is positive, else if the number of positive text is less than that of negative text, the polarity of the test text is negative, when the number of the positive and negative text is equal, we use the algorithm ACCURACY to confirm the polarity of the test text. This algorithm is described in detail as follows.

*Similar k nearest neighbor*

```
Input: the vector of positive text in training set pos_vec, the
vector of negative text in training set neg_vec
Output: the sentiment vector of the text of test set Algorithm
description:
k=0.1;   %set the value of k
```

```
for i=1:Num
  %calculate the number of positive and negative text in training
  %set in the interval
  number_pos = sum(pos_vec(find(pos_vec>=(a_5(i)-k))<=a_5(i)+k));
  number_neg = sum(neg_vec(find(neg_vec>=(a_5(i)-k))<=a_5(i)+k));
  if number_pos-number_neg<0
      a_5(i)=0;
  else if number_pos-number_neg>0
           a_5(i)=1;
       else if a_5(i)<mid_4
               a_5(i)=0;
            else if a_5(i)>=mid_4
                    a_5(i)=1;
```

From the descriptions of these algorithms, we can conclude that the time complexity is $O(n^2)$, that means the algorithms are quick and easy to implement.

## 3 Experiments

We prepare the book review from Amazon.com and test our algorithms. Mutual information is firstly calculated for text features selection. A neural network is used to simulate the function to predict the polarity value of the text.

To select the features using mutual information, we delete those features whose mutual information not greater than 0 and remain 11928 attributes as features at last. To construct a vector, we set the values as 1 if the feature presents in the text, else set values as 0. In experiments, we use a BP neural network, which has one hide layer with 109 neurons, one input layer with the neurons corresponding to the features, and one output layer with one neuron. Sigmoid activation function is employed. We set the output of the training text 0.95 when the orientation of the text is positive, otherwise, the value of the text is 0.05.

We use Matlab6.0 to construct a BP neural network and train the neural network by the training set. Then, we examine the three proposed methods and obtain the turning points.

### 3.1 Data

The book review in Amazon.com contains 2000 texts. In the corpus, there are 1000 texts labeled the positive sentiment and the others are negative texts. We select 1200 texts from the data corpus as training set in which half of them are positive texts and the others are negative one. The remainders are the test data.

### 3.2 Results of Experiments

We compare the midpoint (0.5) and those turning points obtained by our algorithms.

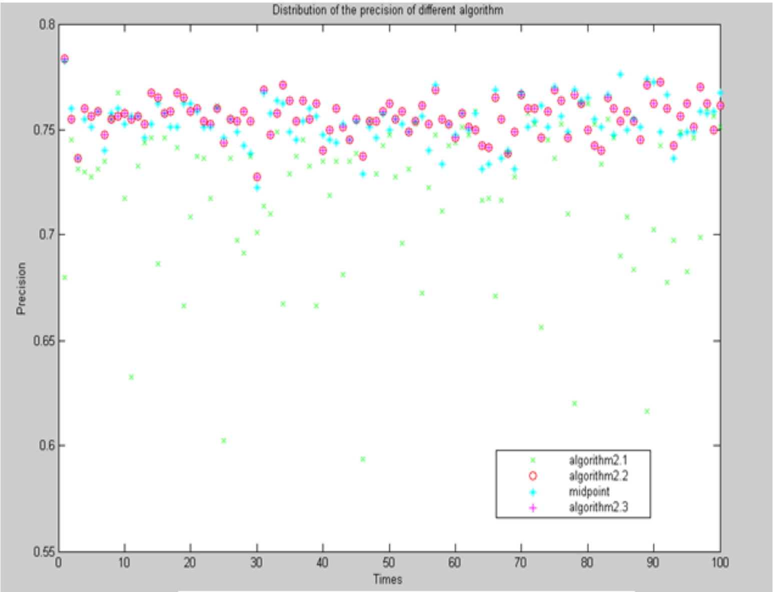


**Fig. 1.** Distribution of the precision of different algorithm

We have trained 100 neural networks, and each neural network has different initial weight. For every neural network, we use the point obtained by different algorithms to determine the polarity of the text separately. The 100 times results of precision using different turning points are shown in Fig.1.

We also calculate the percents of outperformed times than the midpoints. The results are shown in table 1. The turning point values and the average precision for all these algorithms are given in table 2.

We make a compare between the neural network and the Naive Bayesian classifier. The precision of the neural network classifier is 75.34%, and that of the Naive Bayesian Classifier is 66.0401%, lower than the neural network classifier's. We can see that the neural network classifier is better than Naive Bayesian classifier.

**Table 1.** The probability better than mid point

| Algorithm | The probability better than midpoint |
|---|---|
| ADJUST | 9% |
| ACCURACY | 66% |
| NEIGHBOR | 66% |

**Table 2.** The precision of different algorithm

| Algorithm | Turning Point in a neural network | Precision |
|---|---|---|
| Midpoint | 0.5 | 75.34% |
| ADJUST | 0.7317 | 72.17% |
| ACCURACY | 0.47 | 75.59% |
| NEIGHBOR | 0.47 | 75.59% |

From the experiment, we can get some conclusions as follows:

The table 1 shows that, in most time, the turning point got by algorithm ACCURACY and NEIGHBOR, is better than the turning point 0.5, however, the algorithm ADJUST performs worse than the midpoint in the most time.

From the table 2, we can find that, the average precisions of these algorithms are higher than midpoints except the algorithm ADJUST.

We notice that the algorithm ADJUST is the worst in all these algorithms proposed above. The reason is that the basic hypothesis doesn't fit this book review data set. Only for the data set that obey the distribution as the preliminaries of ADJUST described, algorithm ADJUST can get good performance, one of our feature work is to modify something to make the algorithm ADJUST work better.

## 4 Conclusions

In this paper, we showed that, in sentiment analysis, it is not always the best to set the turning point between positive sentiment and negative sentiment by midpoint. Three methods were proposed to get the appropriate value of $x$ as the turning point and the effectiveness of them was examined in experiments. We can make full use of the information contained in training set to improve the precision of the classifier. In the future work, we will modify the methods to better the performance. In addition, it is important to design the appropriate neural network for text sentiment analysis.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China under grant NO.60673043. We also thank Yangsheng Ji and Gang Niu for their instructive discussions related to this work.

## References

1. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. Foundation and Trends in Information Retrieval 2(1-2), 1–135 (2008)
2. Whitelaw, C., Garg, N., Argamon, S.: Using Appraisal Taxonomies for Sentiment Analysis. In: Proceedings of MCLC 2005, the 2nd Midwest Computational Linguistic Colloquium (2005)

3. Whitelaw, C., Garg, N., Argamon, S.: Using Appraisal Groups for Sentiment Analysis. In: Proceedings of the 14th ACM international conference on Information and knowledge management (2005)
4. Yu, B., Kaufmann, S., Diermeier, D.: Exploring the Characteristics of Opinion Expressions for Political Opinion Classification. In: The Proceedings of the 9th Annual International Digital Government Research Conference (2008)
5. Abbasi, A., Chen, H., Salem, A.: Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums. ACM Transactions on Information Systems (TOIS) 26 (2008)
6. Peng, J., Macdonald, C., Ounis, I.: Automatic Document Prior Feature Selection for Web Retrieval. In: Proceedings of the 31st annual international ACM conference (2008)
7. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Proceedings of EMNLP, pp. 79–86 (2002)
8. Shaikh, M.A.M., Prendinger, H., Ishizuka, M.: An Analytical Approach to Assess Sentiment of Text. Computer and information technology (2007)
9. Prabowo, R., Thelwall, M.: Sentiment Analysis: A Combined Approach. Journal of Informetrics 3(2), 143–157 (2009)
10. Han, J., Kamber, M.: Data Mining Concepts and Techniques. China Machine Press (2006)
11. Ruiz, M.E., Srinivasan, P.: Hierarchical Neural Networks for Text Categorization. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (1999)
12. Ruiz, M.E., Srinivasan, P.: Hierarchical Text Categorization Using Neural Networks, vol. 5(1). Springer, Netherlands (2002)
13. Mayer, R.: Text Mining with Adaptive Neural Networks Master Thesis (English), Department of Software Technology and Interactive Systems, Vienna University of Technology (February 2004)
14. Collobert, R., Weston, J.: Fast Sentiment Extraction Using a Novel Neural Network Architecture. In: 45th Annual Meeting of the Association for Computational Linguistics (2007)
15. Shuklin, D.E.: The Structure of a Semantic Neural Network Extracting the Meaning from a Text. Cybernetics and Systems Analysis (2004)

# Research on Natural Disaster Risk Assessment Model Based on Support Vector Machine and Its Application

Junfei Chen, Shihao Zhao, Weihao Liao, and Yuan Weng

State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering,
Business School , Hohai University
No.1 Xi Kang Road, Nanjing, China
chenjunfei@yahoo.com.cn

**Abstract.** The natural disaster risk assessment model based on support vector machine (SVM) is put forward according to the features of natural disaster risk assessment. The indicator system which includes the collapse of houses, the affected areas, the number of casualties, direct economic losses is established by China's actual situation of the regional meteorological disaster. A case for assessing the natural disasters risk of Chinese regions is studied using the established model. The evaluation results show that the evaluation model established is simple and effective. It has good generalization ability in the case of small samples. The results of research in this paper have important reference for natural disaster risk management and decision-making.

**Keywords:** Support Vector Machine, Natural Disasters, Risk Assessment.

## 1 Introduction

Natural disasters are major environmental problems in our society. With population growth and the development of urbanization, natural disasters will be even more severe. According to the information on the International Red Cross, there are roughly 200 major natural disasters each year in the world. These natural disasters caused by an annual average of 130,000 people were killed and 140 million people normal life are affected. Natural disaster risk assessment is the comprehensive reflection of the disaster. Determining the disaster reduction objectives and optimizing the defense measures, evaluating the benefits of disaster reduction are important basis for decision-making [1].

Before 1960, research was mainly limited to natural disasters formation mechanism and disaster prediction, focusing on the investigation and analysis disaster forming conditions and the activities process. The study of various natural disasters has been more widespread attention and more and more quantitative research was done after 1980. Chang established the model based on the socio-economic body for disaster loss and be applied to the water and electricity sectors in Los Angeles by calculating the loss of a more extensive evaluation of earthquake damage to lifeline systems under the resilience of communities [2-3]. Mitchem used principal component analysis to select the natural vulnerability indicators, social vulnerability indicators and

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 762–769, 2009.

capacity indicators for disaster reduction, through the standardization of these indicators and summing to assessment tornado comprehensive vulnerability risk for the venue at the county level in the United State [4]. Disasters assessment has carried out in many areas in China. Water conservancy, agriculture, forestry, meteorology, other departments and a number of experts carried out disaster risk analysis and forecasting evaluation about regional floods, forest disasters and typhoon disasters. The risk map was drawn and many disaster assessment or risk assessment methods and techniques were put forward. Huang put forward a model system for natural disasters risk assessment [5]. Wang used fuzzy clustering analysis to study natural disasters on regional divisions of Sichuan province in China [6]. Feng established an evaluation model to assess the extent of drought based on artificial neural network [7]. Xu and Huang used fuzzy cluster analysis method, BP neural network method to assess the region flood disaster risk [8-9]. Liu assessed the losses caused by strong winds and heavy rain disasters based on grey correlation degree method [10]. Wang established the dynamic risk assessment model under the status of disaster-prone species, multi-body for disaster by drawing in multi-agent modeling to the field of disaster risk assessment and developed new ideas for the field of disaster risk assessment research [11]. Dai established the disaster losses risk evaluation model based on the matter-element analysis method and be applied to classification of natural disaster losses [12].

Support vector machine(SVM) is a new method of machine learning proposed by Vapnik based on statistical theory in the middle of 1990s. Comparing with the traditional neural network learning methods, SVM can effectively solve the model selection and over learning problems, non-linear and dimension disaster, as well as issues such as local minima by solving a quadratic optimization problem to be the global optimal solution. Now, SVM is widely applied in pattern recognition, regression estimation, function approximation, text categorization, time series prediction, financial sequence analysis and so on. It has become research focus problem for experts and scholars which are engaged in machine learning, neural networks, artificial intelligence [13-16].

According to the SVM's property, it is very suitable for comprehensive evaluation. The risk assessment of natural disasters is a typical non-linear, high-dimensional pattern classification problem. The natural disaster risk assessment model based on support vector machine is established and be used in evaluating the region natural disaster risk in China.

## 2 Natural Disaster Risk Management System

Natural disasters mean that people can not control and manipulate disasters which erupt in an instant or short period of time due to the various nature forces outbreak. At present, flood, storm, earthquake, drought is the major natural disasters. Fatal natural disasters refers to the catastrophic events of serious harm to the public life and property due to earthquakes, hurricanes, floods caused by sudden, indifficult to control and avoid. Natural disaster risk management means the whole process of risk management activities including disaster risk management objectives set, identification of disaster risk factors, disaster risk assessment, disaster risk countermeasures, and so on.

Disaster risk management is an important duty of all levels governments and an important aspect of social construction. Faced with the worsening situation of natural disasters, how to strengthen the disaster emergency management system, scientific and effective disposal to deal with natural disasters and better protect public life and property security, promoting economic and social development is significance. The risk assessment of natural disasters is an important aspect of risk management and has an important role for the implementation of effective management of natural disasters.

Through years of practice, the new framework for disaster risk management system has preliminarily established in China. Mainly includeing disaster emergency management organization and command system, emergency management plans for natural disasters, natural disaster risk management early warning disposal system, natural disaster risk management system of material reserves, natural disaster risk management technology support system, natural disaster risk management system of post-disaster reconstruction, and so on.

# 3 Natural Disaster Risk Assessment Model Based on SVM

SVM was first put forward for pattern recognition problem, now SVM has been extended to nonlinear systems of regression estimation, and demonstrates the excellent learning performance. The principle of SVM classification is to find the optimal separating hyper-plane to meet the classification requirements. The actual work of SVM is looking for the generalized optimal separating surface and thus the samples are classified.

## 3.1 C-Support Vector Classification Algorithm

For non-linear classification problem, we can adopt C-support vector classification algorithm. It can divide the data into two categories, the major steps on the algorithm is as follows [17-18].

(1) Suppose the training set of known sample set is $T=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_l,y_l)\}$ . $\boldsymbol{x}_i$ is the input data, $\boldsymbol{x}_i\in R^n$ . $y_i$ is the categories output , $y_i\in\{+1,-1\}$ , $i=1,\cdots,l$ .

(2) Selecting kernel function $K(u,v)$ , construction and solution the following problems :

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)-\sum_{j=1}^{j}\alpha_j \tag{1}$$

$$subject\ to \quad \sum_{i=1}^{l} y_i\alpha_i = 0 \tag{2}$$

$$0\le\alpha_i\le C \quad i=1,2,\cdots,l$$

Where, $C>0$ is the upper bound. According to (1) and (2), we can obtained the optimal solution $\alpha^*$ , $\alpha^*=(\alpha_1{}^*,\alpha_2{}^*,\cdots,\alpha_l{}^*)^T$ .

(3) Calculatiing threshold $b^*$

$$b^* = y_j - \sum_{i=1}^{l} y_i \alpha_i^* K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Where $\alpha_i^*$ is a positive component, $0 < \alpha_i^* < C$.

(4) Construction decision-making function

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i^* K(\boldsymbol{x}, \boldsymbol{x}_i) + b^*\right) \tag{3}$$

At present, several commonly kernel function used are linear kernel function, polynomial function, RBF and sigmoid kernel function, and so on.

### 3.2 Support Vector Machine Multi-classification Algorithm

Natural disaster risk assessment belongs to a multi-classification problem. So, we adopt the multi-classification algorithm to evaluation natural disaster risk. There are mainly including two kind of multi-classification algorithms, that is, one-against-one and one-against-rest. The one-against-one algorithm is superior to the one-against-rest, so we choose the one-against-one algorithm to evaluation natural disaster risk [14,19]. The one-against-one approach in which $k(k-1)/2$ classiers are constructed and each one training data from two different classes. For training data from the ith and the jth classes, we solve the following two-class classication problem:

$$\begin{aligned} \min_{w^{ij}, b^{ij}, \xi^{ij}} \quad & \frac{1}{2}(w^{ij})^T w^{ij} + C\left(\sum_t \xi_t^{ij}\right) \\ \textit{subject to} \quad & (w^{ij})^T \phi(x_t) + b^{ij} \ge 1 - \xi_t^{ij}, \textit{if } x_t \textit{ in the ith class} \\ & (w^{ij})^T \phi(x_t) + b^{ij} \le -1 + \xi_t^{ij}, \textit{if } x_t \textit{ in the jth class} \\ & \xi_t^{ij} \ge 0 \end{aligned} \tag{4}$$

In classication we use a voting strategy: each binary classication is considered to be a voting where votes can be cast for all data points $x$ –in the end point is designated to be in a class with maximum number of votes.

## 4 Case Study

China is the one of the countries of the world's natural disasters most frequent and the loss most serious. With increasing global climate change and extreme weather events, the destructive power of natural disasters and economic losses caused is growing. The floods in Huaihe river basin in 2007, the snow storm and the Wenchuan earthquake disaster in Sichuan province in 2008, serious drought in the north China in early 2009, all this natural disaster has resulted in serious impact on on Chinese economic and social developmt. The lives and property of the public are suffered serious losses.

Therefore natural disasters risk assessment has great significance to China's social and economic sustainable development.

In this paper, China's natural disasters occurred in the region is evaluated based on SVM risk evaluation model. The regional natural disasters mainly include hail, rainstorms, ice storms, typhoons and other natural disasters. The data of natural disasters is adopted in reference [20], as shown in table 1. There are 45 evaluated samples, we select the 35 samples as a training set for support vector machine learning and the results of matter element assessment as the output value. The remaining 10 samples will be as a test set.

**Table 1.** SVM training samples data and training results

| Sample | Region | Disasters | Housing collapse (ten thousands room) | Affected area (ten thousands mu) | number of casualties(one person) | Direct economic losses (one hundred million) | Results of matter-element assessment | SVM training results |
|---|---|---|---|---|---|---|---|---|
| 1 | Hunan | Wind and hailstorms | 20.0 | 50.0 | 1069 | 2.0 | 3 | 3 |
| 2 | Shan dong | Wind and hailstorms | 1.0 | 17.3 | 18 | 1.6 | 1 | 1 |
| 3 | Shan dong | Massive wind and hailstorms | 3.16 | 7.4 | 706 | 1.5 | 2 | 2 |
| 4 | Hunnan | Rainstorm | 30.0 | 2087.0 | 363 | 28.2 | 3 | 3 |
| 5 | Anhui | Rainstorm | 0.08 | 63.0 | 9 | 2.0 | 2 | 2 |
| 6 | Shanxi | Rainstorm | 4.7 | 180.0 | 78 | 8.0 | 2 | 2 |
| 7 | SIchuan | Rainstorm | 25.0 | 55.0 | 16 | 3.5 | 2 | 2 |
| 8 | Gansu | Rainstorm | 0.094 | 67.0 | 21 | 0.158 | 1 | 1 |
| 9 | Shanxi | Ice storm | 2.3 | 440.0 | 52 | 3.0 | 2 | 2 |
| 10 | Anhui | wind and hailstorms | 0.54 | 30.0 | 13 | 2.98 | 2 | 2 |
| 11 | Shanxi | Ice storm | 0.06 | 72.0 | 22 | 1.94 | 2 | 2 |
| 12 | Guanxi | Large rain storm | 0.2 | 42.0 | 6 | 0.847 | 1 | 1 |
| 13 | Shanxi | Large rain storm | 2.0 | 307.0 | 59 | 3.5 | 2 | 2 |
| 14 | Inner Mongolia | Large rain storm | 1.0 | 271.0 | 51 | 0.901 | 2 | 2 |
| 15 | Hebei | Extra-large wind and hailstorms | 1.7 | 1470.0 | 33 | 9.54 | 3 | 3 |
| 16 | Anhui | Large rain storm | 0.4 | 96.0 | 7 | 1.4 | 1 | 1 |
| 17 | Hunan | Rainstorm | 0.9 | 151.0 | 17 | 0.944 | 1 | 1 |
| 18 | Shan dong | Rainstorm | 0.68 | 42.0 | 2 | 0.075 | 1 | 1 |
| 19 | Guan dong | Heavy rainfall of typhoon | 3.4 | 262.0 | 71 | 6.8 | 2 | 2 |
| 20 | Huanan | Typhoon | 6.8 | 1489.0 | 254 | 17.2 | 3 | 3 |
| 21 | Jiangxi | Large rain storm | 15.0 | 600.0 | 15 | 8.0 | 3 | 3 |

**Table 1.** (*continued*)

| Sample | Region | Disasters | Housing collapse (ten thousands room) | Affected area (ten thousands mu) | number of casualties (one person) | Direct economic losses (one hundred million) | Results of matter-element assessment | SVM training results |
|---|---|---|---|---|---|---|---|---|
| 22 | Shanxi | Rainstorm | 11 | 300.0 | 606 | 9.0 | 3 | 3 |
| 23 | Shan dong | Rainstorm | 0.68 | 421.0 | 2 | 0.075 | 1 | 1 |
| 24 | Fujian | Typhoon | 4.4 | 283.0 | 161 | 8.99 | 2 | 2 |
| 25 | Shan dong | Tornado | 9.0 | 750.0 | 164 | 5.0 | 2 | 2 |
| 26 | Sichuan | Large rain storm | 0.7 | 59.0 | 7 | 1.2 | 1 | 1 |
| 27 | Zhe jiang | Typhoon | 4.0 | 665.0 | 65 | 18.0 | 3 | 3 |
| 28 | Jiangsu | Typhoon | 11.5 | 2200.0 | 52 | 20.0 | 3 | 3 |
| 29 | Anhui | Typhoon | 3.0 | 51.0 | 2 | 2.0 | 2 | 2 |
| 30 | Sichuan | Rainstorm | 0.1 | 11.0 | 2 | 0.55 | 1 | 1 |
| 31 | Fujian | Typhoon | 4.8 | 450.0 | 116 | 10.0 | 2 | 2 |
| 32 | Hebei | Rainstorm | 0.22 | 156.0 | 11 | 0.25 | 2 | 1 |
| 33 | Guanxi | Rainstorm | 9.0 | 20.0 | 207 | 0.24 | 2 | 2 |
| 34 | Guan dong | Large rain storm | 0.19 | 44.0 | 10 | 1.2 | 1 | 1 |
| 35 | Guan dong | Rainstorm | 4.19 | 20.0 | 186 | 0.24 | 2 | 2 |

**Table 2.** SVM test samples and test results

| Sample | Region | Disasters | Housing collapse (ten thousands room) | Affected area (ten thousands mu) | number of casualties (one person) | Direct economic losses (one hundred million) | Results of matter-element assessment | SVM test results |
|---|---|---|---|---|---|---|---|---|
| 1 | Fujian | Rainstorm | 1.5 | 216.0 | 69 | 4.1 | 2 | 2 |
| 2 | Hunan | Rainstorm | 2.2 | 319.0 | 37 | 3.3 | 2 | 2 |
| 3 | Hebei | Ice storm | 0.252 | 384.0 | 4 | 1.98 | 2 | 2 |
| 4 | Hainan | Typhoon | 0.12 | 43.0 | 3 | 0.197 | 1 | 1 |
| 5 | Hainan | Large rain storm | 0.19 | 39.0 | 6 | 0.519 | 1 | 1 |
| 6 | Hunan | Large rain storm | 34.0 | 310.0 | 1663 | 28.0 | 3 | 3 |
| 7 | Guanxi | Ice storm | 5.0 | 60.0 | 352 | 0.2 | 1 | 2 |
| 8 | Henan | Wind and hailstorms | 0.87 | 260.0 | 36 | 0.82 | 2 | 2 |
| 9 | Heilong jiang | Tornado | 0.43 | 16.0 | 136 | 0.17 | 1 | 1 |
| 10 | shandong Henan | Rainstorm | 1.8 | 530.0 | 800 | 11.0 | 3 | 3 |

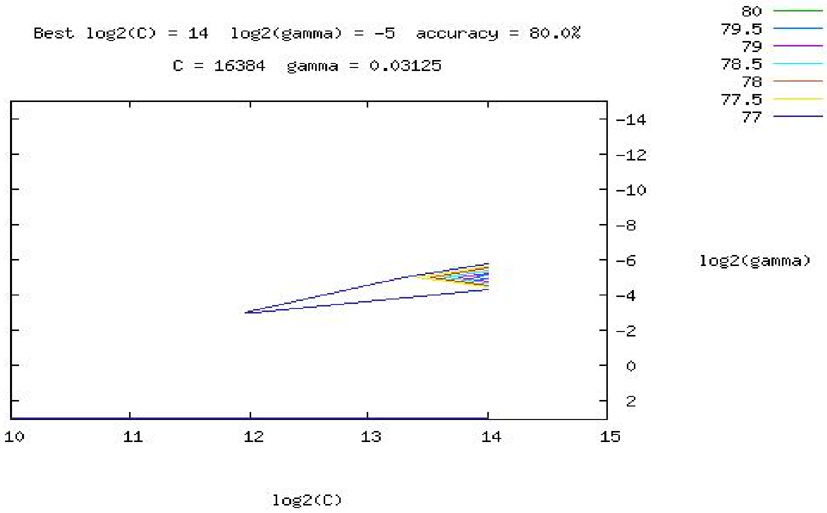


**Fig. 1.** Contour plot of cross validation accuracy

In this paper, the SVM software Libsvm is used to caculation. The Libsvm software has the function including regression calculation and multi-classification pattern recognition based on one-against-one algorithm [19]. The RBF kernel function is selected. The parameter is estimated by cross-validation algotithm. The parameter C equals to 16384.0 and $\gamma$ equals to 0.03125. The algorithm is iterated 107 and the total number of support vector is 17. The contour plot of cross validation accuracy is showed in figure 1.

The training results show that there are 35 training results in 36 samples consistent with matter-element analysis method. The correct rate of training sample output is 97.22%, as shown in table 2. The correct rate of test samples is 90%, as shown in table 3. So, the SVM risk evaluation model in this paper is high-precision and can be used the natural disasters evaluation.

## 5 Conclusion

Natural disasters risk assessment is an important aspect of disaster management. In this paper, the natural disasters risk assessment model based on support vector machine is put forward. The core of this method is to consruct ideal classifier and then inputs the corresponding index data to ideal classifier to evaluation. The method is simple, objective and practical comparing with traditional methods. SVM method can avoid man-made factors when determine the subordinate function and weights of index which the traditional evaluation methods, such as fuzzy comprehensive evaluation and grey clustering methods must determine. SVM can make full use of function of the distribution of training samples and solve the problem of the shortage of small sample data to ensure good generalization performance. SVM is applied to natural disasters risk assessment in this paper. The case shows that the method is feasible and effective. The results of research in this article have important guiding significance for the management departments of natural disaster risk management.

## Acknowledgements

This work was supported in partially by the National Society Science Fund of China(Grant No.09CJY020), Ministry of Water Resources public projects of China (Grant No.200801027), Soft Science Fund of Jiangsu Province (Grant No.BR2008 072), Higher School Society Science Program of Jiangsu Province (Grant No.07 SJD630009),HoHai University Social Science Fund(Grant No.2007425711).

## References

1. Zhang, L., Zhang, Y.C., Luo, Y.H.: Disaster Assessment of Geological Disasters in the Theory and Practice. Geological Publishing House, Beijing (1998)
2. Chang, S.E., Shinozuka, M.: Measuring ImProvements in the Disaster Resilience of Communities. Eartliquake Spectra 20(3), 739–755 (2004)
3. Chang, S.E., Chamberiin, C.: Assessing the Role of Lifeline Systems in Community Disaster Resilience, pp. 87–94. MCEER Publications, New York (2004)
4. Mitchem, J.D.: Place vulnerability to Tornadoes in the United States: A Multi-scale Assessment. University of South, Carolina (2004)
5. Huang, C.F., Shi, P.J.: Two-level Model of Risk Assessment of Natural Disaster on City. Journal of Catastrophology 3(2), 22–27 (1994)
6. Wang, R.Z., Wang, Z.X.: Research on Regional Natural Disasters Division Based on Fuzzy Clustering Analysis: A Case Study in Sichuan Province. Journal of Disaster Science 8(1), 6–12 (1993)
7. Feng, P., Zhang, B.: Assessment of Magnitude of Drought Based on Artificial Neural Network Method. Systems Engineering Theory and Practice 20(3), 141–144 (2000)
8. Xu, H.L., Chen, Y.N.: Flood Hazard Classification Based on Fuzzy Cluster Analysis. Journal of dry Geographical Area 23(4), 350–352 (2000)
9. Huang, T.Z.: Flood Disaster Losses Rapid Assessment Based on BP Neural Network. Journal of Hohai University 31(4), 457–460 (2003)
10. Liu, W.D., Yin, H.B., Cheng, C.L.: Strong Winds and Heavy Rain Disaster Loss Assessment Based on Grey Correlation Degree Method. Meteorological Science and Technology 54(4), 563–566 (2007)
11. Wang, F., Yin, Z.N.: Natural Disasters Dynamic Risk Assessment Model Based on Multi-agent. Geography and Geo-Information Science 25(2), 85–88 (2009)
12. Dai, B.Y., Li, Z.Q., Li, X.L.: A Method of Classing Natural Disaster Loss Based on Matter-element Theory. Journal of Catastrophology 24(1), 1–5 (2009)
13. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
14. Chi-wei, H., Chih-jen, L.: A Comparison of Methods for Multiclass Support Vector Machine. IEEE Transacatutions on Neural Networks 13(2), 415–425 (2002)
15. Gold, C., Sollich, P.: Model Selection for Support Vector Machine Classification. Neurocomputing 55, 221–249 (2003)
16. Smola, A.J., Schölkopf, B.: A Tutorial on Support Vector Regression. Statistics and Computing 14, 199–222 (2004)
17. Deng, N.Y.: New Data Mining Method: Support Vector Machine. Science Press, Beijing (2004)
18. Xiao, J.H., Wu, J.P., Yang, S.Z.: Research on Comprehensive Evaluation Method Based on Support Vector Machine. Journal of Computer Engineering 28(8), 28–30 (2002)
19. LIBSVM—A Library for Support Vector Machines (2005), http://www.kernel-machines.org
20. Li, Z.Y., Deng, X.M.: Natural Disasters Evaluation Based on Matter Element Analysis Model. Journal of Natural Disasters 3(2), 28–33 (1994)

# Identifying Tobacco Control Policy Drivers: A Neural Network Approach

Xiaojiang Ding[1], Susan Bedingfield[1], Chung-Hsing Yeh[1],
Ron Borland[2], David Young[2], Sonja Petrovic-Lazarevic[3], and Ken Coghill[3]

[1] Clayton School of Information Technology, Monash University, Victoria, Australia
[2] VicHealth Centre for Tobacco Control, The Cancer Council Victoria, Australia
[3] Department of Management, Monash University, Victoria, Australia
{Xiaojiang.Ding,Sue.Bedingfield,
Chunghsing.Yeh}@infotech.monash.edu.au

**Abstract.** This paper presents a neural network approach to investigating Australian smokers' quit motivations that affect their quit attempts. Based on the data from the International Tobacco Control Four Country Survey, Neural network (NN) models are developed to identify smokers' quitting motivations as smokers' quit motivations are significant factors in predicting smokers' quit attempts. In order to identify the underlying tobacco control policies from these quitting motivations, principle component analysis is used to group individual attributes into 4 tobacco control policy drivers: Personal Concerns, Price, Social Restrictions and Social Encouragement which are related to specific tobacco control policies. To examine the impact of these tobacco control policy drivers on smokers' making a quit attempt, a set of NN models using 4 policy drivers are also built. Experimental results indicate that in comparison with cigarette price and social restrictions, educating smokers of health benefits from quitting and social encouragement for cessation of smoking have more impacts in encouraging smokers to make a quit attempt.

**Keywords:** neural networks, quitting motivations.

## 1 Introduction

With the increasing concerns regarding the negative effects of tobacco consumption, global effort has been made to investigate the relationship between tobacco control policies and smokers' quitting behaviours [1]. The Framework Convention on Tobacco Control (FCTC), which is the first international public health treaty, recalls the global-wide co-operations to be involved in the devotion of tobacco control [1][2]. Under FCTC, the conduction of the International Tobacco Control (ITC) Policy Evaluation Project aims at analyzing the impact of national level of tobacco control policies via the survey data of adult smokers from four developed countries: the United States, Canada, the United Kingdom and Australia [2]. Recent research has shown that smokers' quitting motivations have been identified as significant factors in predicting smokers' quitting intentions [3][4]. Common smokers' motivations for thinking about quitting include their personal health, cigarette price,

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 770–776, 2009.

family influence and advice from professionals. These factors are highly correlated with tobacco control policies, thus investigating smokers' quitting motivations can be regarded as significant guidelines for proposal or modification of relevant tobacco control policies.

Statistics techniques have been applied in recent research studies [5][6]. However, these studies have limitations in dealing with complex data. The aim of this study is to identify tobacco control policy drivers for investigating Australian smokers' quitting motivations. In this study, we first construct NN models to predict smokers' quit attempts. We then use principal component analysis to group attributes for policy-related evaluations and discuss the results of this study.

## 2 The International Tobacco Control Survey Data

The data for this study are from Australian adult smokers who completed all the questions in the first year (2002) of ITC Four Country Survey (ITC-4C), a cohort study which provides evidence base for ITC project (Thompson et al., 2006). The data from this survey is called Wave 1 data. Altogether 1,513 respondents from Australia were collected, among these 532 (35.16%) smokers made a quit attempt (MQA). In this study, we use MQA as the output variable of NN models to predict Australian smokers' quit attempts, as MQA is an important step leading to quitting cessation behaviours [3][8]. We use 12 questions from ITC-4C survey to assess Australian smokers' motivations for thinking about quitting. Table 1 shows the survey questions.

**Table 1.** Questions to assess smokers' motivations for thinking about quitting

| **Even though you mentioned that you are not currently planning to quit, in the past 6 months, have each of the following things led you to think about quitting, not at all, somewhat, very much?**<br>**1 – Not at all**<br>**2 – Somewhat**<br>**3 – very much** | |
|---|---|
| **BQ01** | Concern for your personal health? |
| **BQ02** | Concern about the effect of your cigarette smoke on non-smokers? |
| **BQ03** | Society disapproves of smoking? |
| **BQ04** | Price of cigarettes? |
| **BQ05** | Smoking restrictions at work? |
| **BQ06** | Smoking restrictions in public places like [restrictions OR bars/cafes OR pubs]? |
| **BQ07** | Advice from doctor, dentist, or other health professional to quit? |
| **BQ08** | Free or lower cost, stop-smoking medication? |
| **BQ09** | Availability of telephone helpline/Quitline/information line? |
| **BQ10** | Advertisements or information about the health risks of smoking? |
| **BQ11** | Warning labels on cigarette packages? |
| **BQ12** | Setting an example for children? |

The data used in this study are the answers to these 12 questions and each answer is on a three-point scale: 1 = not at all, 2 = somewhat and 3 = very much.

## 3 Building Neural Network Models Using Motivation Attributes

Neural network models are widely applied to model complex data and examine the relationship between input and output variables [7]. In this study, we use back-propagation NN models to predict smokers' quit attempts. All 12 attributes listed in Table 1 are used as input variables and MQA is the output variable. In the NN models, the activation function used is logistic and the number of hidden neurons is the sum of the number of input neurons and the number of output neurons. Sample data are equally split into a training set and a test set. We first load the training set into the NN model, then we test the accuracy of the test set. One advantage of NN is that for each particular problem, instead of using default output, we can adjust the decision threshold to achieve a desired level of accuracy [9]. Experiments are conducted by varying the decision threshold. We identify that when the decision threshold is 0.33, the accuracies of training set and test set achieve the best results, which are 72.03% and 70.85%, respectively. In this NN model, both the training set and the test set accuracies are close, thus indicating the generalisation of the model built. In addition, it implies that smokers' quitting motivations identified in the ITC-4C survey can be regarded as strong predictors in predicting their quit attempts. However, individual attributes are difficult to represent the underlying tobacco control policies. To facilitate the research of corresponding policies from these individual attributes shown in Table 1, a set of policy drivers needs to be identified.

## 4 Policy Drivers

Policy drivers can be described as a group of attributes that can be used to develop a policy for increasing the rate of smokers' MQA. The identification of policy drivers can help to examine the effects of tobacco control policies. As such, we try to identify the links between tobacco control policies and smokers' quitting motivations. Principle component analysis (PCA) is widely used to identify new meaningful underlying factors based on given attributes [10]. In this study, we use PCA to group attributes into factors. The statistical package SPSS16.0 is used for conducting PCA experiments. We use the default values in parameter settings except selecting 'varimax' in Rotation method option. We also set the suppress absolute value to less than 0.3. Table 2 shows the result of rotated component matrix using PCA when the 12 attributes in Table 1 are loaded. According to the derived matrix, we examine the common theme for each factor and provide a label representing a policy driver. As shown in Table 2, three factors have been clearly generated except that BQ04 remains ambiguous as the loading values (0.474 and 0.504) are close. When examining this attribute, the question in relation to this attribute is about cigarette price which has no clear correlation with other attributes. As a result, we identify this attribute as a new factor to represent a policy driver.

**Table 2.** Rotated component matrix using PCA for 12 attributes

| | Factors | | |
|---|---|---|---|
| | **1** | **2** | **3** |
| **BQ01** | | | 0.711 |
| **BQ02** | | | 0.711 |
| **BQ03** | | 0.632 | 0.312 |
| **BQ04** | | 0.474 | 0.501 |
| **BQ05** | | 0.777 | |
| **BQ06** | | 0.771 | |
| **BQ07** | 0.464 | | |
| **BQ08** | 0.562 | | |
| **BQ09** | 0.788 | | |
| **BQ10** | 0.691 | | 0.369 |
| **BQ11** | 0.687 | | |
| **BQ12** | 0.310 | | 0.604 |

Based on the PCA, 12 attributes are grouped into 4 factors as policy drivers, referred to as Personal Concerns, Price, Social Restrictions and Social Encouragement as shown in Table 3.

**Table 3.** Tobacco control policy drivers and corresponding attributes

| Policy Drivers | | | |
|---|---|---|---|
| **Personal Concerns** | **Price** | **Social Restrictions** | **Social Encouragement** |
| BQ01<br>BQ02<br>BQ12 | BQ04 | BQ03<br>BQ05<br>BQ06 | BQ07<br>BQ08<br>BQ09<br>BQ10<br>BQ11 |

In this study, for each policy driver we use the average value of its associated attributes. The number of the possible value is dependent on the number of attributes. For Personal Concerns and Social Restrictions, there are 7 values for both policy drivers: {1, 1.33, 1.67, 2, 2.33, 2.67, 3}; for Price, there are 3 values: {1, 2, 3}; and for Social Encouragement, 11 values are derived based on 5 attributes in this policy driver: {1, 1.2, 1.4, 1.6, 1.8 2, 2.2, 2.4, 2.6, 2.8 3}

## 5 Building Neural Network Models Using Policy Drivers

In order to evaluate the effects of the policy drivers derived in Table 3, experiments are conducted. First, we develop a set of NN models for predicting smokers' quit attempts. The input variables are four policy drivers from Table 3 and the output variable is smokers' MQA. The parameter settings in the NN models are the same as the ones in our previous experiments. The data set is also equally split into a training set and a test set to check the robustness of the model based on the accuracies generated. Then by adjusting the decision thresholds, we identify the best accuracies for both the training set and the test set which are 68.58% and 67.90%. Table 4 shows the confusion matrix of the accuracy of the test set on the selected model.

**Table 4.** Confusion matrix of classification accuracies of the test set

| Class | Actual MQA | Actual NMQA | | |
|---|---|---|---|---|
| **Classified MQA** | 184 | 215 | 399 | 46.12% |
| **Classified NMQA** | 87 | 270 | 357 | 75.13% |
| | 271 | 485 | 756 | |
| | 67.90% | 55.67% | | |

Table 5 shows the comparison of accuracies using different input variables in our NN models. The accuracies between the two models are fairly close indicating that using the 4 policy drivers as inputs to the NN model has similar prediction accuracies as using the original 12 attributes. Thus, our finding suggests that the derived 4 policy drivers can fully represent the original 12 attributes based on the performance of NN models built.

**Table 5.** Accuracies using different inputs in the NN models

| Inputs of the NN models | Training set | Test set |
|---|---|---|
| **12 Attributes (Table 1)** | 72.03% | 70.85% |
| **4 Policy Drivers (Table 3)** | 68.58% | 67.90% |

As smoking behaviour is difficult to change, policy decision makers need to assess which factors may significantly influence smokers' to think as well as actually making a quit attempt, and ultimate quitting. Table 6 shows the weight list of four factors in the NN model generated. It shows that Personal Concerns has the highest weight, thus indicating that Personal Concerns is the most significant factor in the NN prediction model. Social Encouragement ranks second which contributes 29.92% in constructing the NN model. Social Restrictions is in the third place and Social Encouragement has the least contribution.

**Table 6.** Weight percentage of each factor in the NN model using 4 policy drivers as inputs

| Policy Drivers | Weight (%) |
|---|---|
| **Health Benefit** | 46.91 |
| **Price** | 9.11 |
| **Social Restrictions** | 14.07 |
| **Social Encouragement** | 29.92 |

Findings from this study suggest that Australian smokers are more likely to make a quit attempt if they are more concerned about their personal health and quitting advices from professionals. These findings have important implications for designing or strengthening relevant tobacco control policies which include propagandizing the knowledge of quitting benefits for both smokers and their family's health, adding more quit lines and information lines, distributing low or free stop-smoking medications and enlarging health warning on cigarette packs. These findings are consistent with our previous research studies [8][11]. Compared to personal concerns about health and social encouragement, cigarette price and social restrictions have less impact in encouraging smokers to make a quit attempt.

## 6 Conclusion

In this paper, we have presented a neural network approach to investigating Australian smokers' quit motivations as well as providing suggestions in designing relevant tobacco control policies. To examine whether smokers' quitting motivations have a strong impact in predicting smokers' quit attempts, we have built a set of NN models and selected the best model by varying the decision thresholds. Then we have applied the PCA to group all the quitting motivations into 4 policy drivers which have been identified to have high correlations with the tobacco control policies. It has demonstrated that the derived 4 policy drivers can fully represent the original 12 input attributes according to the prediction accuracies of the NN models. Experimental results have shown that personal concerns about health and social encouragement have significant impacts in encouraging Australian smokers to make a quit attempt.

## References

1. Fong, G.T., Cummings, K.M., Borland, R., Hastings, G.B., Hyland, P., Giovino, G.A., Hammond, D., Thompson, M.E.: The conceptual framework of the International Tobacco Control (ITC) Policy Evaluation Project. Tobacco Control 15(3), 3–11 (2005)
2. Thompson, M.E., Fong, G.T., Hammond, D., Boudreau, C., Driezen, P., Hyland, P., Borland, R., Cummings, K.M., Hastings, G.B., Siahpush, M., Machintosh, A.M., Laux, F.L.: Methods of the International Tobacco Control (ITC) Four Country Survey. Tobacco Control 15(3), 12–18 (2006)
3. Hyland, A., Borland, R., Li, Q., Yong, H.-H., McNeill, A., Fong, G.T., O'Connor, R.J., Cummings, K.M.: Individual-Level Predictors of Cessation Behaviours among Participants in the International Tobacco Control (ITC) Four Country Survey. Tobacco Control 15(3), 83–94 (2006)
4. Yun, C.J., Ding, X., Bedingfield, S., Yeh, C.-H., Borland, R., Young, D., Petrovic-Lazarevic, S., Coghill, K., Zhang, J.Y.: Performance Evaluation of Intelligent Prediction Models on Smokers' Quitting Behaviour. In: Fyfe, C., Kim, D., Lee, S.-Y., Yin, H. (eds.) IDEAL 2008. LNCS, vol. 5326, pp. 210–216. Springer, Heidelberg (2008)
5. Yong, H.-H., Borland, R., Siahpush, M.: Quitting-Related Beliefs, Intentions, and Motivations of Older Smokers in Four Countries: Findings from the International Tobacco Control Policy Evaluation Survey. Addictive Behaviors 30, 777–788 (2005)

6. Zhou, X., Nonnemaker, J., Sherrill, B., Gilsenan, A.W., Coste, F., West, R.: Attempts to quit smoking and relapse: Factors associated with success or failure from the ATTEMPT cohort study. Addictive Behaviors 34, 365–373 (2009)
7. Yeh, C.-H., Lin, Y.-C.: Neural Network Models for Transforming Consumer Perception into Product Form Design. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3973, pp. 799–804. Springer, Heidelberg (2006)
8. Ding, X., Bedingfield, S., Yeh, C.-H., Borland, R., Young, D., Zhang, J.Y., Petrovic-Lazarevic, S., Coghill, K.: A Decision Tree Approach for Predicting Smokers' Quit Intentions. Journal of Electronic Science and Technology 6(3), 284–288 (2008)
9. Smith, K.A.: Introduction to Neural Networks and Data Mining for Business Applications. Eruditions Publishing, Melbourne (1999)
10. Jolliffe, I.T.: Principal Component Analysis. Springer, New York (1986)
11. Ding, X., Bedingfield, S., Yeh, C.-H., Borland, R., Young, D., Petrovic-Lazarevic, S., Coghill, K., Zhang, J.Y.: A rule based approach to modeling the effect of tobacco control policies. In: Proceedings of the 4th International Conference on Information and Automation for Sustainability (ICIAfS 2008), Colombo, Sri Lanka, pp. 496–501 (2008)

# Intrusion Detection Using Neural Networks: A Grid Computing Based Data Mining Approach

Marcello Castellano[1,2], Giuseppe Mastronardi[1,2], and Gianfranco Tarricone[1]

[1] Dipartimento di Elettrotecnica ed Elettronica Politecnico di Bari
via Orabona, 4 70125 – Bari – Italy
castellano@poliba.it

[2] e.B.I.S. s.r.l. (electronic Business in Security), Spin-Off of Polytechnic of Bari,
Via Pavoncelli, 139 – 70125 Bari – Italy

**Abstract.** Scientific disciplines such as life sciences as well as security and business fields depend on Knowledge Discovery because of the increasing amount of data being collected and for the complex analyses that need to be performed on them. New techniques, such as parallel, distributed, and grid-based data mining, are often able to overcome some of the characteristics of current data sources such as their large scale, high dimensionality, heterogeneity, and distributed nature. In several of these data mining applications, neural networks can be successfully applied. Moreover, an approach using neural networks seems to be one of the most promising methods for intrusion detection in a computer system or network security today. In this paper we describe a grid computing data mining approach for an intrusion detection application based on neural networks. Detection is carried out through the analyses of internet traffic generated by users in a network computer system.

**Keywords:** Knowledge Discovery, Data Mining, Neural Networks, Intrusion Detection, Grid Computing.

## 1 Introduction

Knowledge discovery is a complex and interdisciplinary field which deals with the understanding of unsuspected patterns and rules in data. The discovering of these relationships is a highly interactive process where, in order to achieve satisfying results, the user must be allowed to apply various techniques or given parameters on a permanent basis [1].

A number of techniques using different approaches have been proposed with the aim of developing a reliable Knowledge Discovery Process. Recently, Mining techniques are considered to be the most effective strategy for Knowledge Discovery.

Data Mining involves finding patterns in data which can be considered interesting, according to certain user-defined interestingness measures, and valid, based on user defined validation measures. Industry has also become more and more interested in this area of research since data mining can be useful in analyzing and understanding

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 777–785, 2009.

large collections of data in different formats, including data warehouses, databases, file archives, digital libraries. The increasing trend towards decentralized business organizations, user distribution, software and hardware systems combined with the characteristics of today's data sources, such as their large volume, high dimensionality, heterogeneity, and distributed nature has call for more advanced and flexible approaches and solutions. Useful data mining techniques in these contexts are: parallel data mining which deals with tightly-coupled systems based on fast networks and highly centralized platforms; distributed data mining which deals with loosely-coupled systems with unpredictable interconnection latencies; grid-based data mining which comprises some characteristics of the above two fields, as well as more specific issues and requirements like the management of multi-institutional virtual organizations [2,3].

Neural networks have been successfully applied in several of these data mining applications and have become one of the most promising methods for intrusion detection in a computer system as in network security.

In security, it is axiomatic that what cannot be prevented, must be detected. The goal of Intrusion Detection is to identify security violations in computer systems. Intrusion Detection is a passive approach to security as it monitors information systems and sounds alarms when security violations are detected. Examples of violations include the abuse of privileges or the use of attacks to exploit software or protocol vulnerabilities [4,5]. Various approaches for recognizing a possible intrusion have been presented in the literature. For example, the use of interval timers and event counters to build time-series models have been proposed. This can generate statistical models used to compare short-term behaviour with long-term historical norms based on temporal regularities in audit data. By forming user profiles, web usage patterns analysis from Web log files can also provide a better understanding of user behaviour on an individual or group basis. Approaches based on audit data are widely used to build intrusion detection models. In these methods, data must be labelled as either "normal" or "attack" in order to define suitable behavioural models to represent these two different categories [6,7,8,9].

This paper presents an architectural approach based on grid data mining which used neural networks to carry out intrusion detection.

## 2 Knowledge Discovery in Databases

The field of Knowledge Discovery in Databases (KDD) consists of a new generation of techniques and tools to analyze intelligently and automatically large volumes of data or 'mines' in order to 'extract' hidden knowledge. Knowledge Discovery in Databases is the process of identifying regularity (patterns) and characteristic patterns in the data as well as patterns which have a certain degree of general validity and are not discounted or otherwise known, but potentially useful and easily understood [1]. KDD process consists of the following steps: *Selection and Sampling; Pre-processing and Cleaning; Transformation and Reduction; Data Mining (Choice of algorithm, Identification of parameters, Processing, Evaluation of the model); Evaluation and Visualization of results.*

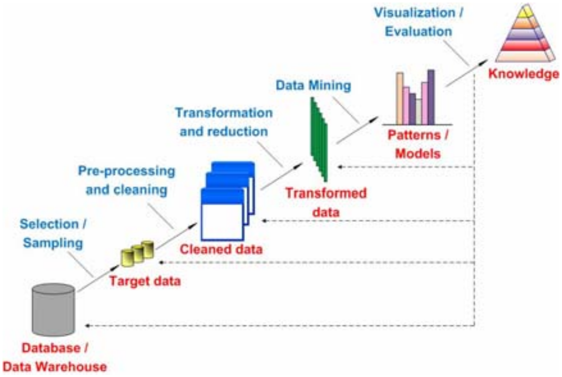


**Fig. 1.** Steps for Knowledge Discovery in Databases

## 3 Parallel, Distributed and Grid Data Mining

Data mining is a massive computing task that deals with memory resident data. Traditional data mining techniques are often inadequate for the stored data in centralized or distributed systems. Hence, the need for parallel and distributed computing becomes necessary when handling large-scale data mining [10,11].

Parallel Data Mining (PDM) is targeted for tightly-coupled systems, like shared or distributed memory machines, communication via high speed buses, shared file systems, and clusters based on fast or dedicated networks. A PDM algorithm shares the entire database and constructs a global model. Distributed Data Mining (DDM) deals with loosely-coupled systems: clusters with average-fast or slow networks and geographically distributed computing nodes. The main differences between PDM and DDM are the number of computing nodes involved, the communication costs, and the degree of data distribution. The goal of DDM is to perform data mining operations by using the peculiarities and the availability of distributed resources [11]. A DDM algorithm partitions the database at several sites and mines a local model for each partition. Then, it merges these to build a global model. Grid Data Mining (GDM) shares many commonalities with PDM and DDM.

Problems encountered by applications based on the mining methods described so far often involve different aspects of the data such as their large volume, high dimensionality, heterogeneity, and distributed nature. If these problems are addressed by traditional machines that sequentially work on an entire collection of homogeneous and independent data, any limitations encountered will probably be technological. An economic solution has dynamic scaling characteristics that adapt to the size of the data collection being analyzed and is offered by systems weakly linked to calculating networks. Recently, a computational paradigm is being explored which suggests creating computer technology pools. The use of this technology is called Grid Computing.

Grid technology offers a configurable environment and can potentially provide applications with an architecture for easy and transparent access to geographically distributed heterogeneous resources, like data storage, networks, and computational resources across different organizations and administrative domains [12,13,14].

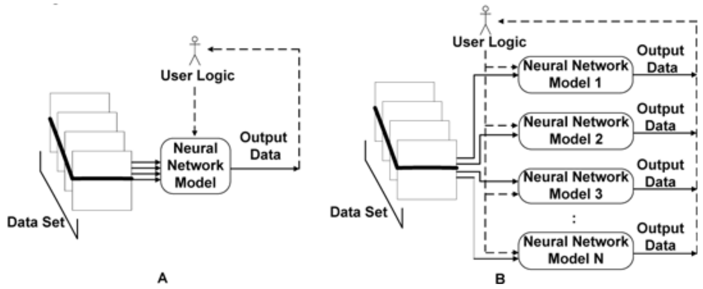


**Fig. 2.** An MISD application through the Grid

This work presents an MISD application of Grid computing. The prototype was developed to simplify the creation and distribution of a MISD grid enabled application. In addition, it was designed to integrate and to manage its MISD applications within the system using its modular structure. By using the grid application, the user should be able to move away from a situation like the one illustrated in Figure 2.A and into one as described in Figure 2.B.

Since learning for bigger networks with large training sets is time consuming, it is crucial to look at parallel implementation schemes to reduce the training time required. Different types of parallelism possible for backpropagation neural networks are: *Training session parallelism*, which starts training sessions with different initial training parameters on different processing elements (PE); *Training set parallelism*, which splits the training set across the PEs. Each element has a local copy of the complete weight matrix and accumulates weight change values for the given training patterns; *Pipelining*, which pipelines the training patterns between the layers, i.e. compute hidden and output layer on different processors. The output layer processor can calculate output and error values for the present training pattern while the hidden layer processor works on the next training pattern. The forward and backward phase may also be parallelized in a pipeline; *Node parallelism*, which computes the neurons within a layer in parallel (named neuron parallelism). In addition, computation within each neuron may also run in parallel (named weight or synapse parallelism) [15].

A common problem is choosing the best neural network model for a predefined input data set. Grid-based architecture can support this selection since it allows different models of neural networks on the grid nodes to be installed. In addition, it can reduce the time needed for selecting the best network model through a parallel decision-making process. The paper attempts to illustrate how to make this choice by comparing the results obtained from many backpropagation network models. The neural network models were trained and validated in a Matlab environment using Neural Network Toolbox.

## 4 Experimental Results

The architectural model described here was used to produce a system prototype which could detect a possible intrusion. Detection was carried out through the analysis of internet traffic generated by users in a network computer system. An intruder can be defined as a user able to illegitimately overcome login phase, and then surf the internet in an unauthorized manner using access credentials which are not his own. Since a user will normally generate web traffic based on his own habits and preferences, a user profile was created using data traffic analysis. The presence of an intruder could then be detected by comparing the traffic analysis with the user profile previously created.

The prototype created for this intrusion detection system was based on Knowledge Discovery techniques. The system implementation was divided into several phases. In the first phase, useful data was collected to create a user model which could then be analyzed. In this phase, a log file of the user's internet browsing habits was created. Next, a user profile was produced on the basis of the log file analysis. At this point, the user model was ready to be employed for intrusion detection. If, by analyzing the user log, an intrusion was detected, a new user model was created from this analysis. If the user was legitimate, the more recent model should be similar to the original model. Instead, if the model was not similar to the original profile within a specific tolerance, an intrusion could be suspected. Figure 3 shows a conceptual scheme of the system.

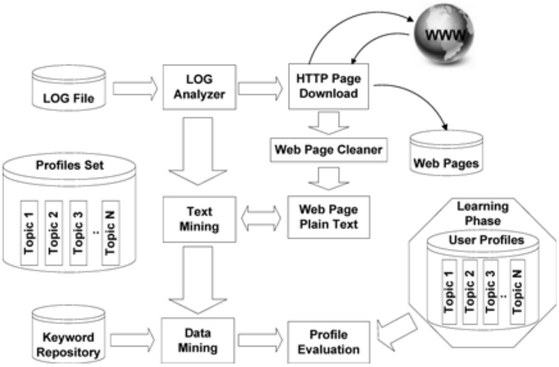


**Fig. 3.** Logic of the Intrusion Detection prototype

Thus, the identification of an intrusion using a backpropagation neural network was also explored in this work. The transfer function of hidden neurons can be described by the sigmoid function:

$$g(x,\theta) = (1+ e^{-\beta(x-\theta)})^{-1} \tag{1}$$

where $\theta$ and $\beta$ are the neuron threshold and the gain factor, respectively.

The synaptic matrix $W$ was trained using the backpropagation techniques in the *incremental learning* mode: the weights were updated after each pattern $X$ was forwarded to the NN inputs, as follows:

$$\Delta w_{i,k} = -\eta\ \partial E[W] / \partial w_{i,k} + \alpha(\Delta w_{i,k}{}^{(OLD)}) \tag{2}$$

where $E[W]$ is the distance between net output $Y(X)$ and the known expected answers $T(X)$. The *momentum term,* $\alpha\Delta w_{i,k}^{(old)}$ is introduced in such a way that each connection value $w_{i,j}$ tends to change in the *average downhill* direction, avoiding sudden oscillations.

The performance evaluation of the best neural network model was approached by using a multilayered NN with one hidden layer and $N_h = 2N_i$ hidden neurons [16,17]. The input layer consists of $N_i = 5$ neurons which represented the number of topics on the analyzed web page. The output layer consisted of $N_o = 2$ elementary processing units which each identified a different class of user. An output value of 1.0 and 0.0 in the two output nodes was expected when the analysis indicated normal user whereas a value of 0.0 and 1.0 in the two output nodes indicated a possible intrusion. The network parameters used for a suitable implementation of the gradient descent algorithm are: learning rate $\eta = 1$, momentum $\alpha = 0.9$ and gain factor $\beta = 1$ [18]. The neural network began with an analysis of internet traffic and, in particular, it examined the occurrence in percentage of a series of topics inside the log files analyzed. It produced a detection system which then made the final decision about the user's activities indicating either 'normal' or 'intrusion' behavior. For each input, the network produced an output which was then compared to the desired output (target). For each epoch, a test data set was input into the network and the mean square error (mse) was evaluated. The training phase stopped when the overtraining effect [18] was observed and the best synaptic matrix was frozen. In this study, the synaptic matrix was frozen at the 16$^{th}$ epoch, as can be seen in Figure 4.

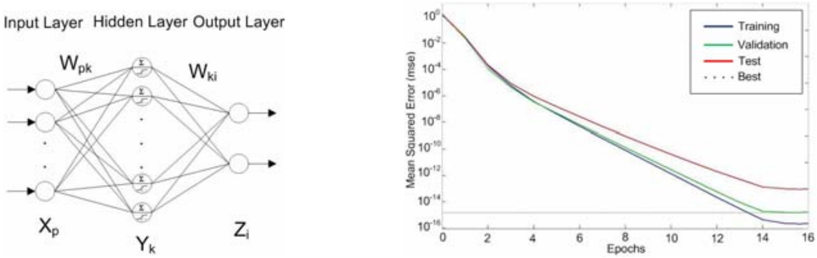


**Fig. 4.** Backpropagation neural network and rate of convergence in the training phase (blue line) and test phase (red line) in comparison with the validation curve (green line). The overtraining effect occurs at the 16th epoch.

Figure 5 shows a graphical representation of a user's profile generated by the system.

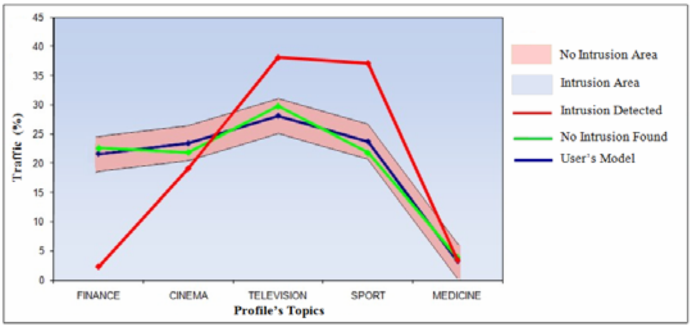


**Fig. 5.** Results from the log analysis, illustrating cases of authorized and unauthorized navigation

The profile is represented with a broken line. This line connects the percentages associated with the profile topics. The broken line is surrounded by a tolerance band. The tolerance level, represented by this band width, was defined by the system administrator. An intrusion could then be recognized graphically when a profile obtained from a subsequent analysis differed more than the fixed tolerance level allowed. In the chart, the red line indicating the intruder's user profile can be seen to exit the area of the tolerance band.

System efficiency was evaluated by considering the execution time estimated on a serial computer and then that on a computational grid. Execution time was estimated for a grid with a variable nodes number considering an input data set composed of 1,000 log files. An estimation of a speedup indicator was obtained. The speedup indicator was defined as the ratio between the mining process execution time on a traditional serial machine and that on a computational grid.

First, a run-time module of the input data set, which consisted of 1,000 log files, was evaluated on a traditional serial machine as follows:

$$T_{\text{sequential\_time}} = T_{\text{initialization}} + T_{\text{one\_log file\_execution\_time}} \bullet N_{\text{log files}} \tag{3}$$

Considering a typical ADSL bandwidth and the user program module run-time on the one node, the total run-time on the grid was:

$$T_{\text{grid\_time}} = T_{\text{initialization}} + (T_{\text{upload\_time}} + T_{\text{download\_time}}) \bullet N_{\text{log files}} + T_{\text{upload\_code}} \bullet \\ \bullet\ N_{\text{nodes}} + (T_{\text{one\_log file\_execution\_time}} \bullet N_{\text{log file}}) / N_{\text{nodes}} \tag{4}$$

Let:

$$S = T_{\text{sequential\_time}} / T_{\text{grid\_time}} \tag{5}$$

The speedup factor referred to how much faster a parallel algorithm could be in comparison to a corresponding sequential algorithm in using the grid. The factor was computed for a node number ranging from 1 to 60 for various size data sets. Figure 6 shows the trend of the speedup coefficient in correspondence to the growth of the node multiplicity.

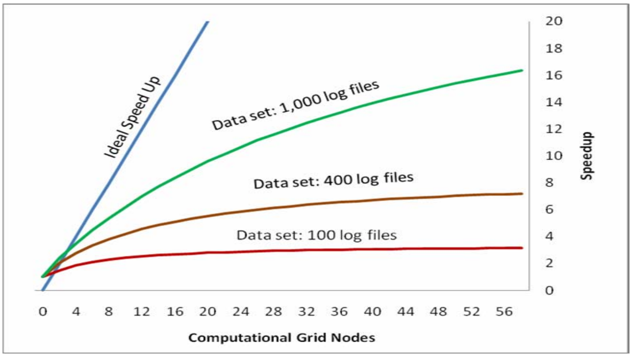


**Fig. 6.** Speedup depending on grid node number, considering a data set ranging from 100 to 1,000 log files

## 5 Conclusion

This paper has discussed how to develop an Intrusion Detection solution using neural networks based on a grid computing Data Mining approach. This solution was based on a Knowledge Discovery process using grid-based data mining technologies. A prototype able to detect a possible intrusion by the analysis of the internet traffic generated by users of a network computer system has been presented. Finally, the results of tests conducted on a backpropagation neural network and an evaluation of the speedup factor of the grid have been discussed.

**Acknowledgments.** The authors acknowledge the financial support provided by the Italian Ministry of Education, University and Research and by the Strategic Project "Research and Development of a Maintenance Management System of Railway Assets (ASSET)", which have made possible the realization of this work as result of our research activities.

## References

1. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R.: Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, London (1996)
2. Lee, W., Stolfo, S.J., Mok, K.W.: Data mining approaches for intrusion detection. In: 7th USENIX Security Symposium, San Antonio, TX (1998)
3. HajHmida, M.B., Congiusta, A.: Parallel, Distributed, and Grid-Based Data Mining: Algorithms, Systems, and Applications. In: Cannataro, M. (ed.) Handbook of Research on Computational Grid Technologies for Life Sciences, Biomedicine and Healthcare. Medical Information Science Reference, pp. 90–119 (2009)
4. Kemmerer, R.A., Vigna, G.: Intrusion Detection: A Brief History and Overview. Part supplement IEEE Computer 35(4), 27–30 (2002)
5. Tront, J.G., Marchany, R.C.: Internet security: intrusion detection & prevention. In: Proc. of the 37th Annual Hawaii International Conference on System Sciences, vol. 1 (2004)
6. Lee, W., Stolfo, S.J., Mok, K.W.: Mining Audit Data to Build ID Model. In: 4th International Conference on Knowledge Discovery and Data Mining, pp. 66–72 (1998)
7. Julisch, K.: Data mining for Intrusion Detection: a Critical Review. In: Barbara, D., Jajodia, S. (eds.) Applications of Data Mining in Computer Security. Kluwer Academic Publisher, Dordrecht (2002)
8. Pepyne, D.L., Hu, J., Gong, W.: User Profiling for Computer Security. In: American Conference on Control, Boston, pp. 982–987 (2004)
9. Esposito, M., Mazzariello, C., Oliviero, F., Romano, S.P., Sansone, C.: Real Time Detection of Novel Attacks by Means of Data Mining. In: ACM ICEIS Conference (2005)
10. Freitas, A.A.: A survey of parallel data mining. In: 2nd International Conference on the Practical Applications of Knowledge Discovery and Data Mining, pp. 287–300. The Practical Application Company, London (1998)
11. Kargupta, H., Chan, P.: Advances in Distributed and Parallel Knowledge Discovery. MIT Press, Cambridge (2000)
12. Reed, D.A., Mendes, C.L., Lu, C., Foster, I., Kesselmann, C.: The Grid 2: Blueprint for a New Computing Infrastructure – Application Tuning and Adaptation. Morgan Kaufmann, San Francisco (2003)

13. Foster, I., Kesselmann, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan-Kaufmann edition, San Francisco (1998)
14. Allcock, B., Bester, J., Bresnahan, J., Chervenak, A.L., Foster, I., Kesselman, C., Meder, S., Nefedova, V., Quesnal, D., Tuecke, S.: Data Management and Transfer in High Performance Computational Grid Environments. Parallel Computing Journal 28(5) (2002)
15. Torresen, J., Landsverk, O.: A Review of Parallel Implementations of Backpropagation Neural Networks. In: Sundararajan, N., Saratchandran, P. (eds.) Parallel Architectures for Artificial Neural Networks: Paradigms and Implementations, pp. 25–64. Wiley-IEEE Computer Society Press, Chichester (1998)
16. RumelHart, D.E., McClelland, J.L.: Parallel Distributed Processing, vol. 1, p. 318. MIT Press, Cambridge (1986)
17. Kolmogorov, A.N.: Dokl. Akad, Nauk. USSR 114, p. 953 (1957)
18. Pao, Y.H.: Adaptive Pattern Recognition and Neural Networks. Addison-Wesley, Reading (1989)

# Recurrent Neural Networks as Local Models for Time Series Prediction

Aymen Cherif[1], Hubert Cardot[1], and Romuald Boné[1,2]

[1] Université François Rabelais Tours
Laboratoire d'informatique
64 avenue Jean Portalis, 37200 Tours, France
[2] Ecole Nationale d'Ingénieurs
du Val de Loire
Rue de la Chocolaterie, BP 3410, 41034 Blois cedex, France

**Abstract.** Local Models for regression have focused a great deal of attention in recent years. They have proved to be more efficient than global models and especially when dealing with chaotic time series. Many models have been proposed to cluster time series and they have been combined with several predictors. In this paper we present an extension for recurrent neural networks in allowing to apply them to local models and we discuss the obtained results.

## 1 Introduction

Time series forecasting is a widely discussed problem which can be found in many disciplines. Researchers from various disciplines (finance, mathematics, physics, machine learning, etc.) have addressed this question. With this emphasis, many models and algorithms have been proposed, starting with statistical methods, Multi-Layer Perceptron (MLP [1]) and Recurrent Neural Networks (RNN [1]). In addition to these methods, many researchers have proposed methods and techniques to push the limits of prediction error. The "local models" [2,3,4] technique has been well used in the last decade. It consists in dividing the data into homogeneous clusters using Vector Quantization (VQ [5]) to simplify the prediction task on each cluster and most of the time inspired from the Self-Organizing Maps algorithm (SOM [6]).

Since RNN have demonstrated in many occasions better results than other methods and specially for chaotic time series [7], we propose in this paper a method to use the RNN as a local model. In the second section, a brief description of the previous work on the local approach is presented. In the third section, we will describe the method used in this paper. Then some tests are presented and discussed. Eventually, we conclude with some perspectives.

## 2 Vector Quantization and Forecasting

In global models, the learning process consist in presenting the entire learning set to the unique predictor. The inputs are formed as a temporal window $X(t)$

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 786–793, 2009.

defined by the equation:

$$X(t) = (x(t), x(t-\gamma), ...., x(t-(T-1)*\gamma)), \tag{1}$$

with $T \in \mathbb{N}$ the temporal window width and $\gamma$ the time step. In general, global predictor can be described as a function to estimate $\hat{f}$ defined as:

$$\hat{x}(t) = \hat{f}(X(t)), \tag{2}$$

where $\hat{x}(t)$ is the predicted value at time $t$. However, the performance of a single global model can be quickly limited with chaotic or seasonal time series [8,4,9]. On the other hand, the local model operates in two successive phase. During the first one, the time series is embedded into a $M$-dimensional space of vectors similar to equation 1, then clustered into sub learning sets. Dividing the data into clusters should allow the predictor to estimate a more simple function in each cluster. The clustering is generally done using Vector Quantization technique.

In practice, the most used algorithm for local modeling is the Self-Organizing Maps (SOM [6]). It can be directly applied to create different classes of predictors corresponding to each cluster. According to the literature, several choices can be made. In [2,4] the regressors used were linear, while in [10] non linear regressors (Radial Basis Function Network) were used. It is possible to modify the SOM input vectors to include the future value $(x(t+1))$ [4,3,11]. And finally it is possible to choose between the result from the predictor of the winning cluster [12,2], or a combination of predictors [13,14,10].

However, since the original Self-Organizing Maps does not take into account the temporal sequences processing, many researchers have proposed extensions to fill in this gap. A first idea is to use a recurrent processing of time signal with recurrent winner computation. The first modification proposed is the Temporal Kohonen Maps (TKM [15]), in which an activation value is calculated and accumulated for each output unit map to determine the winner node. Then the recurrent SOM (RSOM [12,16] uses the same idea, replacing the scalar activation value by an accumulated vector. Recursive SOM (RecSOM [17]) uses recursively the original SOM algorithm. The Sequential Activation, Retention and Delay Network SARDNET [18] algorithm adds a deactivation mechanism to the prototypes. The Feedback SOM (FSOM [19]) uses a feedback layer for the recurrent information processing.

A second approach consists in mapping the temporal dependencies to spatial correlation. In this category we can cite the SOM with Temporal Activity Diffusion (SOMTAD [20]) which propagate the information in its node structure and the Merge SOM (MSOM [21]) which introduce a contextual information in the prototype (context vector provided by the previous time step) in addition to weight vector.

At present, the common point in all previously mentioned works is the use of predictors for each obtained cluster. In the first proposition, the time series were supposed to be locally linear [2], that is why linear models such as AR predictors were used. But when dealing with more complex and chaotic data, these models were replaced by non linear ones [12] (i.e. MLP). Since the superiority of the RNN has been proved in several works [7], we introduce here a simple modification to this RNN allowing to use them as local models in combination with a simple SOM algorithm.

## 3 RNN with Local Approach

In Recurrent Neural Networks, the connections between units form a directed cycle. Particular configurations of RNNs exist such as the Jordan architecture [22] and the Elman architecture [23]. Both have feedback connections with the context layer. More general configurations are also possible, for example the fully connected neural network and the fully connected hidden layer. The RNN must be approached differently than feed-forward networks. In fact, in the latter, the information is propagated linearly from layer to layer. The RNN uses also information coming from later processing stages (due to cycles). A well-used algorithm is the Back-Propagation Through Time (BPTT [1]). Its main idea is to unfold in time the original RNN to obtain an MLP with l layers (l is the number of steps). When applying BPPT on the learning set between $t_1$ and $t_l$, we obtain the following expression for the variance of one weight $\Delta w_{ij}(t_1, t_l - 1)$:

$$\Delta w_{ij}(t_1, t_l - 1) = \sum_{t_1}^{t_l - 1} \Delta w_{ij}(\tau) = -\eta \sum_{t_1}^{t_l - 1} \frac{\partial E(t_1, t_l)}{\partial w_{ij}(\tau)} \tag{3}$$

$E(t_1, t_l)$ being the mean square quadratic error and $w_{ij}(\tau)$ the copy of $w_{ij}$ for $t = \tau$ in the unfolded network BPTT employs. If we note $s_j(t)$ the output neuron $j$ at time $t$ we can write:

$$\frac{\partial E(t_1, t_l)}{\partial w_{ij}(\tau)} = \frac{\partial E(t_1, t_l)}{\partial net_i(\tau)} s_j(\tau). \tag{4}$$

It should be noted that after Vector Quantization the sequential order of presented vector is no longer assured. This is not a problem for the MLP as local model which has only to solve a regression problem given $X(t)$ from equation (1) a vector from a particular cluster and containing $T$ past values. In the case of RNN, the internal feedback creates an implicit memory that contributes in the prediction in addition to the network inputs. In this case presenting the data in a sequential order is obligatory for correct predictions. For this reason we propose to present on each of our local model the entire learning set to which we added extra information about the cluster membership, $\delta_{t,k}$, defined as:

$$\delta_{t,k} = \begin{cases} 1 & \text{if the vector } X(t) \in C_k \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where $C_k$ denotes the cluster number $k$.

Therefore, the final learning rule (of eq 4) is provided in equations (6) and (7) of the $k^{th}$ RNN, and are updated in order to take into account only the vectors belonging to the cluster $C_k$. The $\delta_{t,k}$ is introduced in equations (6) and (7) for that aim:

- for $\tau = t_l - 1$

$$\frac{\partial E(t_1,t_l)}{\partial net_i(\tau)} = \begin{cases} (s_i(\tau) - d_i(\tau))\delta_{t,k} f_i^{'}(net_i(\tau)) & \text{if } i \in O(\tau) \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

- for $t_1 \leq \tau \leq t_l - 1$

$$\frac{\partial E(t_1,t_l)}{\partial net_i(\tau)} = \begin{cases} \left[ (s_i(\tau) - d_i(\tau)) + \displaystyle\sum_{j \in succ(i)} \frac{\partial E(t_1,t_l)}{\partial net_j(\tau+1)} w_{ij}(\tau+1) \right] \delta_{t,k} f_i^{'}(net_i(\tau)) \\ \text{if } i \in O(\tau) \\ \displaystyle\sum_{j \in succ(i)} \frac{\partial E(t_1,t_l)}{\partial net_j(\tau+1)} w_{ij}(\tau+1)\delta_{i,k} f_i^{'}(net_i(\tau)) \\ \text{otherwise,} \end{cases} \tag{7}$$

where $O(\tau)$ is the set of output neurons at time $\tau$ and $d_i(\tau)$ the desired output at time $\tau$.

As a result, the BPTT algorithm will operate normally when $X(t) \in C_k$. On the contrary, the information is just propagated inside the network with no gradient calculation, but allowing it to be used in future stage, which is the principle of the RNN.

## 4 Experimental Results

In this section we tested the proposed method on well known time series (Mackey-Glass and FIR-Laser). For comparison purposes we went back to earlier works [9,4,8,2,12] which have used the same data under similar conditions. In order to see clearly the contribution of the RNN on the results we used the original SOM algorithm for the present tests.To evaluate the results we used the Normalized Mean Square Error (NMSE) criteria.

### Experimentation on Mackey-Glass Time Series

The Mackey-Glass (MG) time series can be generated from the MG delay differential equation [24]:

$$\dot{x}(t) = \frac{\alpha x(t-\tau)}{1 + x^c(t-\tau)} - \beta x(t), \tag{8}$$

We generated 5000 points using the $4^{th}$ order Runge-Kutta method. The first 3000 points were used in the learning set, while the remaining 2000 points were divided between the validation and the testing set.

Figure (1) shows the data repartition between clusters after execution of the SOM algorithm on the MG17 data and with one dimensional output layer, on

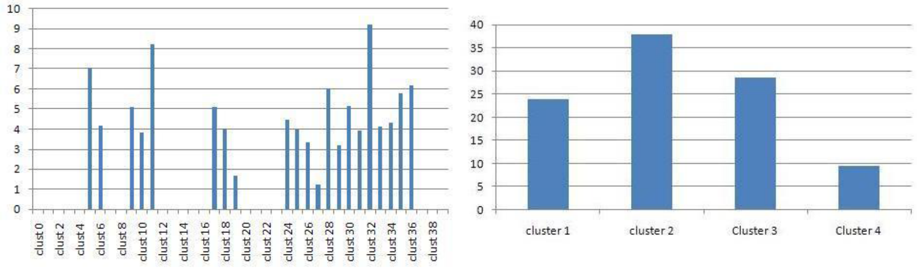


**Fig. 1.** Repartition of data between clusters. Left part, 40 units on the SOM map, Right part, 4 units on the SOM map.

the right part 4 units were used, on the left part 40 units were used. We chose 1-D layer instead of 2-D layer to do the tests because they demonstrated a better convergence stability in the learning process, they required a less processing time and we imagine that they are more representative for time series signal.

For the 4 units case, we remark that each time one or two clusters were the most represented (Figure 1, right part). The risk in this case is to obtain in the end non homogeneous clusters as wished and then to influence the prediction error. However, when using a larger number of units, we remark the creation of regions of activated units in the map (Figure 1, left part) due to the topological function in the SOM learning phase [6] (a gaussian function in our case). With these results we can be more confident in the clusters homogeneity, but with a more important computation time since we use the whole learning set for each RNN predictor on each cluster.

Table (1) compares the results of our method with those obtained in other previous works. In [4] a SOM with a 35 × 35 output layer with local MLP predictors were used. In [8] Dynamic Cell Structure combined with Local Linear Models were used (DCS-LLM). In [7] a RNN with 7 neurons in the hidden layer was used (this is the same network used as local model). The results show that the RNN as local model (SOM+RNN) outperforms the other methods.

In Table (1), we can see that in a single step prediction ($\gamma = 1$), our approach ameliorates the performance of a global RNN (without delay) dividing

**Table 1.** Results for the MG-17 task, showing (from left to right) the NMSE $\times 10^3$ for a single time step ($\gamma = 1$), 6 steps ahead ($\gamma = 6$)

| Method | MG-17 NMSE | |
|---|---|---|
| | $\gamma = 1$ | $\gamma = 6$ |
| SOM+MLP (35x35 units) [4] | – | 4.8 |
| DCS-LLM [8] | – | 5.5 |
| RNN [7] | 0.235 | – |
| SOM+RNN (4 units) | 0.0512 | 2.5 |

approximatively the error by 10 when using only 4 clusters in the SOM. We also remark that with different prediction steps ($\gamma = 6$), our approach obtains better results that other local models (0.0025 instead of 0.0048 for [4]).

The contribution of the local approach can be seen clearly seen by analyzing the results of each cluster separately. For example in the case of four units in the SOM Layer, we reach a very small prediction error in certain clusters $(2,93 \times 10^{-5}; 1,04 \times 10^{-4}; 5,55 \times 10^{-6}; 2,47 \times 10^{-6})$.

### Experimentation on Santa Fe Laser Time Series

The laser data is a set from the Santa Fee time series prediction competition [25]. It consists of measurements of the intensity of an infrared laser in a chaotic state. The training set contains 901 samples, 552 are used for the validation and 201 are used for testing.

The results listed in Table (2) are compared with the results obtained in the other related works. In [27] Recurrent SOM was used. In [28] a mixture of predictions and true values as input (Error Propagation, EP) was used. In [25] a non linear regression approach was used in a maximum likelihood framework realized with feed-forward NN (12 neurons on the hidden layer). As global model we present the results of FIR MLP [29] and a RNN [7] as global model with 7 neurons in the hidden layer and one neuron in the input layer. This same NN is used as local model in our approach, combined with a SOM. We present the results with 4 SOM output units and 40 SOM output units.

Our conclusion for the table (2) is that the local model prediction in FIR-Laser time series is more complicated than MG-17 (described above) mainly due to a smaller number of samples in the learning set. Although the number of used cluster ameliorates the results (from $0,093$ to $0,0205$), the best performance still in the global model [7] ($0,00792$).

The analysis of the prediction error in each clusters of the 40 units example, only 6 clusters in the total of 40 have obtained a prediction error bigger than [25] which is the best results from the table 2. We suggest as future work that choosing an adapted clustering algorithm for temporal sequence can be a possible solution in this case to profit more of local models.

**Table 2.** Results for the FIR-Laser task, showing the NMSE for a single time step ($\gamma = 1$)

| Method | FIR-Laser NMSE |
|---|---|
| FIR MLP [26] | $0,023$ |
| RSOM [27] | 0.0833 |
| Weigend [25] | 0.016 |
| Bakker [28] | 0.02159 |
| RNN [7] | 0.00792 |
| SOM+RNN (4 units) | 0.092 |
| SOM+RNN (40 units) | 0.0205 |

## 5 Conclusion

In this paper we have proposed to use a RNN with an adaptation of BPTT as local model using Vector Quantization techniques (SOM in our case). We compared our results with other local approach methods and we found that these results are encouraging for further studies. We preferred to use the original SOM algorithm in order to clearly see the contribution of RNN as local model. However, we saw that a better clustering could have a positive impact on the final performances. This is the main reason that leads us to think that it could be interesting to use other more adapted algorithms for temporal sequences clustering (see section 2). A disadvantage in this method is the important learning time, which is due to the use of the whole learning set for all predictors. For this problem we can consider as a possible solution a second clustering of the units, which can provide a compromise between learning time and performance with, at the same time, a reduced number of clusters and a reasonable homogeneity of samples inside a cluster.

## References

1. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. Parallel distributed processing: explorations in the microstructure of cognition: Foundations 1, 318–362 (1986)
2. Walter, J., Riter, H., Schulten, K.: Nonlinear prediction with self-organizing maps. In: International Joint Conference on Neural Networks (IJCNN), vol. 1, pp. 589–594 (1990)
3. Simon, G., Lee, J.A., Cottrell, M., Verleysen, M.: Forecasting the CATS benchmark with the Double Vector Quantization method. Neurocomputing 70(13-15), 2400–2409 (2007)
4. Vesanto, J.: Using the som and local models in time-series prediction. Technical report, Helsinki University of Technology (1997)
5. Gray, R.M., Neuhoff, D.L.: Quantization. IEEE Transactions on Information Theory 44(6), 2325–2383 (1998)
6. Kohonen, T.: Self-organization and associative memory, 3rd edn. Springer, New York (1989)
7. Boné, R.: Recurrent Neural Networks For Time Series Forecasting. PhD thesis, Université de Tours, Tours, FRANCE (2000)
8. Chudy, L., Farkas, I.: Prediction of chaotic time-series using dynamic cell structuresand local linear models. Neural Network World 8, 481–489 (1998)
9. Gers, F.A., Eck, D., Schmidhuber, J.: Applying LSTM to Time Series Predictable through Time-Window Approaches. In: Dorffner, G., Bischof, H., Hornik, K. (eds.) ICANN 2001. LNCS, vol. 2130, pp. 669–676. Springer, Heidelberg (2001)
10. Simon, G.: Méthodes non linéaires pour séries temporelles: prédiction par Double Quantification Vectorielle et sélection du délai en hautes dimensions. PhD thesis, FSA/ELEC - Département d'électricité (2006)
11. Barreto, G.A., Araujo, A.F.R.: Identification and control of dynamical systems using the self-organizing map. IEEE Transactions on Neural Networks 15(5), 1244–1259 (2004)

12. Varsta, M., Millán, J.D.R., Heikkonen, J.: A recurrent self-organizing map for temporal sequence processing. In: Gerstner, W., Hasler, M., Germond, A., Nicoud, J.-D. (eds.) ICANN 1997. LNCS, vol. 1327, pp. 421–426. Springer, Heidelberg (1997)
13. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation 3(1), 79–87 (1991)
14. Cottrell, M., Girard, B., Rousset, P.: Long term forecasting by combining Kohonen algorithm and standard prevision. In: Gerstner, W., Hasler, M., Germond, A., Nicoud, J.-D. (eds.) ICANN 1997. LNCS, vol. 1327, pp. 993–998. Springer, Heidelberg (1997)
15. Chappell, G.J., Taylor, J.G.: The temporal kohonen map. Neural Networks 6(3), 441–445 (1993)
16. Varsta, M., Heikkonen, J., Lampinen, J., Del Millán, J.R.: Temporal kohonen map and the recurrent self-organizing map: Analytical and experimental comparison. Neural Processing Letters 13(3), 237–251 (2001)
17. Voegtlin, T.: Recursive self-organizing maps. Neural Networks 15(8-9), 979–991 (2002)
18. James, D.L., Miikkulainen, R.: Sardnet: A self-organizing feature map for sequences. In: Advances in Neural Information Processing Systems, vol. 7, pp. 577–584. MIT Press, Cambridge (1995)
19. Mizushima, F., Toyoshima, T.: Language learnability by feedback self-organizing maps. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) ICONIP 2006. LNCS, vol. 4234, pp. 228–236. Springer, Heidelberg (2006)
20. Euliano, N.R., Principe, J.C.: A Spatio-Temporal Memory Based on SOMs with Activity Diffusion. In: Oja (ed.) Kohonen Maps, pp. 253–266. Elsevier, Amsterdam (1999)
21. Strickert, M., Hammer, B.: Merge som for temporal data. Neurocomputing 64, 39–71 (2005); Trends in Neurocomputing: 12th European Symposium on Artificial Neural Networks (2004)
22. Jordan, M.I.: Serial order: A parallel distributed processing approach. Technical report, California Univsity, San Diego, La Jolla Institute for Cognitive Science (1986)
23. Elman, J.L.: Finding structure in time. Cognitive Science 14(2), 179–211 (1990)
24. Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control systems. Science 197(4300), 287–289 (1977)
25. Weigend, A.S., Nix, D.A.: Prediction with confidence intervals (local error bars). In: Proceedings of internaltional Conference on Neural Information Processing (ICONIP 1994), pp. 847–852 (1994)
26. Eric Wan, A.: Finite Impulse Response Neural Networks for Autoregressive Time Series Prediction. PhD thesis, Stanford (1993)
27. Koskela, T., Varsta, M., Heikkonen, J., Kaski, K.: Recurrent SOM with local linear models in time series prediction. In: 6th european Symposium on Artificial Neural Networks. ESANN 1998, pp. 167–172 (1998)
28. Bakker, R., Schouten, J.C., Giles, C.L., Takens, F., van den Bleek, C.M.: Learning chaotic attractor by neural networks. Neural Computation 12 (2000)
29. Wan, E.A.: Finite impulse response neural networks for autoregressive time series prediction. In: Proceeding in the NATO Advanced Workshop on Time Series Prediction and Analysis (1993)

# Construction of the General Physical Condition Judgments System Using Acceleration Plethysmogram Analysis Results

Heizo Tokutaka[1], Eikou Gonda[2], Yoshio Maniwa[3], Masashi Yamamoto[4],
Toshiyuki Kakihara[5], Masahumi Kurata[5], Kikuo Fujimura[5],
Li Shigang[5], and Masaaki Ohkita[1]

[1] SOM Japan Inc.
[2] Yonago National College of Technology
[3] Yabu City Minamidani Clinic
[4] Shikano-Onsen Hospital
[5] Tottori University
4-637, Koyamacho-kita Tottori, Japan
tokuhema@hal.ne.jp
http://www.somj.com

**Abstract.** Among the popular lifestyle-related diseases are smoking, overweight and stress. A daily health check is important because there is no clear objective symptom for these diseases. We developed diagnostic software which shows the state of the blood vessels using a Basic SOM model, and compared 1map (Basic SOM) with Torus and a Spherical SOM (blossom).

**Keywords:** plethysmogram, SOM, clinical case study.

## 1 Introduction

To carry oxygen and nutrition all over the body is a blood vessel's most important task. When healthy, the blood vessel has the elasticity and flexibility corresponding to the pressure of the blood. When dangerous factors such as smoking, overweight, stress and other lifestyle-related diseases influence a blood vessel, arteriosclerosis gets worse. It causes cerebral infarction and cerebral hemorrhages due to a high blood pressure. These accidents can happen suddenly without clear symptoms, so that a daily check is becoming increasingly important for prevention.

From the fingertip, the acceleration plethysmogram is obtained without subjective stress in a short period of time. We developed diagnostic software which graphically shows the state of the blood vessel by using a Basic SOM model. Much information is included in the plethysmogram about the blood movement in the vessel, which goes from the center (heart) to the end (fingertips). The bloodstream travels through the blood vessels from the heart to the capillary vessels in a wave-like motion. The plethysmograph is affected by the ictus cordis,

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 794–801, 2009.

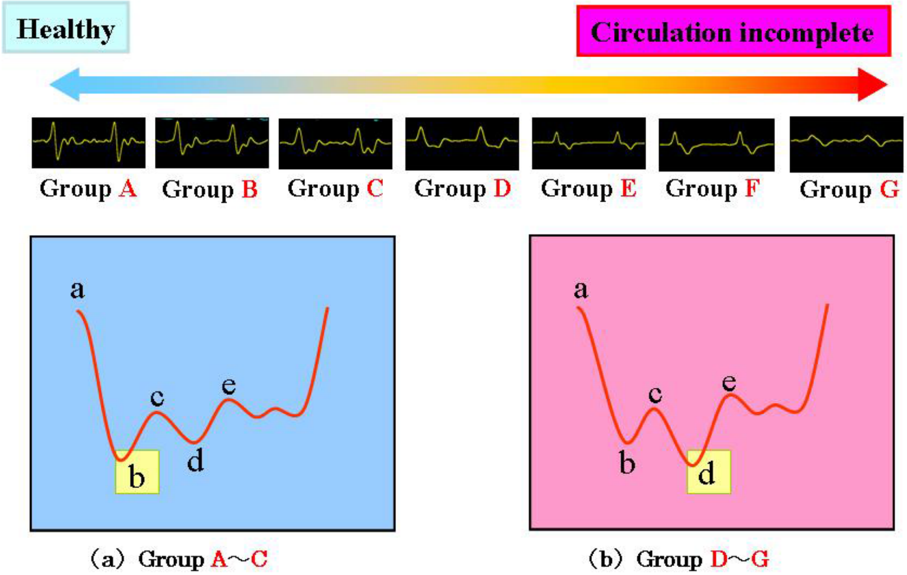


**Fig. 1.** The features of the acceleration plethysmogram. The group (a) in the right hand side show healthier waves than the one (b).

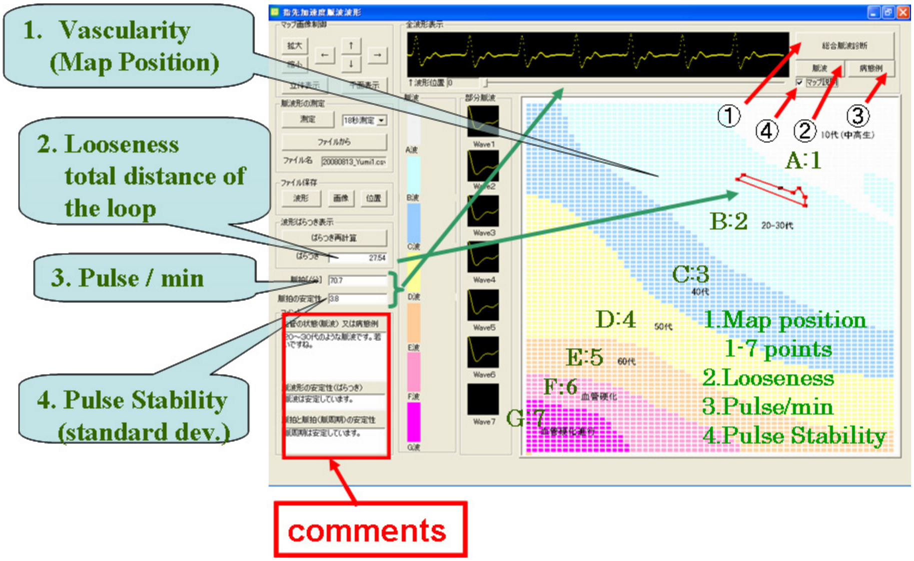
the haemodynamics and the physiological condition caused by the change in the properties of an arteriole. The effects can be observed as distortions in the wave profiles. The inside diameter of the blood vessel changes due to a swelling of the blood vessel. The wave motion that occurs at that time is called the volume plethysmogram. The volume plethysmogram has the problem that the base line never becomes stable. Therefore, it is difficult to estimate the inflection point because the wave becomes sparse when it raises. Hence, techniques to differentiate the waveform have been proposed. The acceleration plethysmogram (2nd derivative) is one recent example that has considered and evaluated [1]. The plethysmogram used for the diagnosis at present is the acceleration plethysmogram, and a doctor is evaluating the plethysmogram by watching the location of the inflection point or by calculating a blood vessel age formula.

The waveforms of the acceleration plethysmogram and the presently used characteristics (or features) are shown in Fig. 1. Waveforms are typically categorized into seven classes, as illustrated in Fig. 1. The figure shows the gradual changes from the waveform of a healthy signal (Group A) to a possibly unhealthy waveform (Group G), which could be caused by an incomplete blood circulation [1]. When the labels "a" to "e" are put on the wave extremal points, "b" is smallest for a healthy subject and "d" smallest when an incomplete circulation is possible (refer to Fig. 1). Doctors and researchers can assess the state of the blood vessel from the waveforms, but a non-specifically trained person. The plethysmogram analysis software described below has been developed using a Self Organizing Maps (SOM) [2,3] so that a non-trained person may interpret the waveform. The conventional tool is calculating the vein age by Eq. (1), which the medical doctor [4] has developed using the peak values of a , b, c, d, e shown in Fig. 1 (almost all of the commercial tools work this way). It is developed in such a way that our tool Pulser-SOM [5,6] becomes more easy to use, even by the general public. Thus, our tool Pulser-SOM is developed for the general public for easily assessing the condition of their capillary vessels [5,6].

$$[Vein\ age] = 43.5 \times \frac{(b - c - d - e)}{a} + 65.9 \tag{1}$$

## 2 Synthetic Plethysmogram Diagnosis

Pulser_SOM is an application which obtains the state of the blood stream from the plethysmogram. The waveform of the acceleration plethysmogram, to differentiate the volume plethysmogram, obtained from the fingertip, has the characteristic of predicting the state of the blood stream. This characteristic cannot be understood by the non-trained individual, only by doctors and researchers. However, a non-trained person, who doesn't have the special knowledge, can



↑波形位置 0

ファイル名 20080813_Yumi1.csv

ばらつき 27.54

脈拍[/分] 70.7

脈拍の安定性 3.8

血管の状態(脈波) 又は病態例

20~30代のような脈波です。若いですね。

脈波形の安定性(ばらつき)

脈波は安定しています。

脈拍と脈拍(脈周期)の安定性

脈周期は安定しています。

**Fig. 2.** The general pulse wave analysis tool in which the pulse wave analysis and the clinical example can be seen on the same screen. At present, an acceleration Plethysmogram pulse wave is analyzed, by pushing the button "the pulse wave" of ②. Then, the analysis can be moved to the general pulse wave analysis by pushing the button ① and using the data of 1 - 4 in the figure. Putting the mark in the button ④, the Japanese explanation of each color region on the map of an acceleration Plethysmogram pulse wave analysis mode appears.

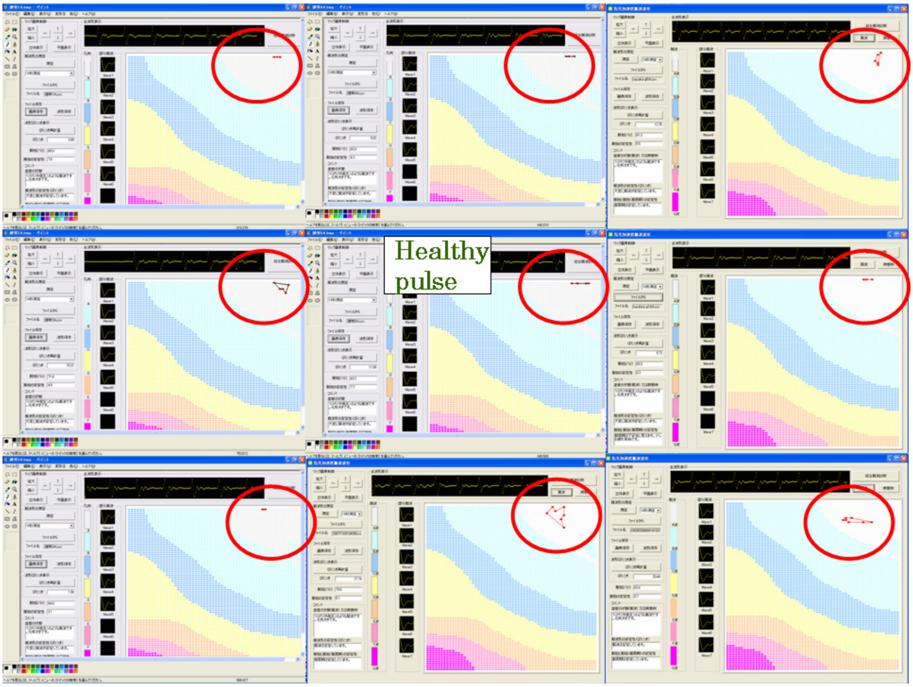


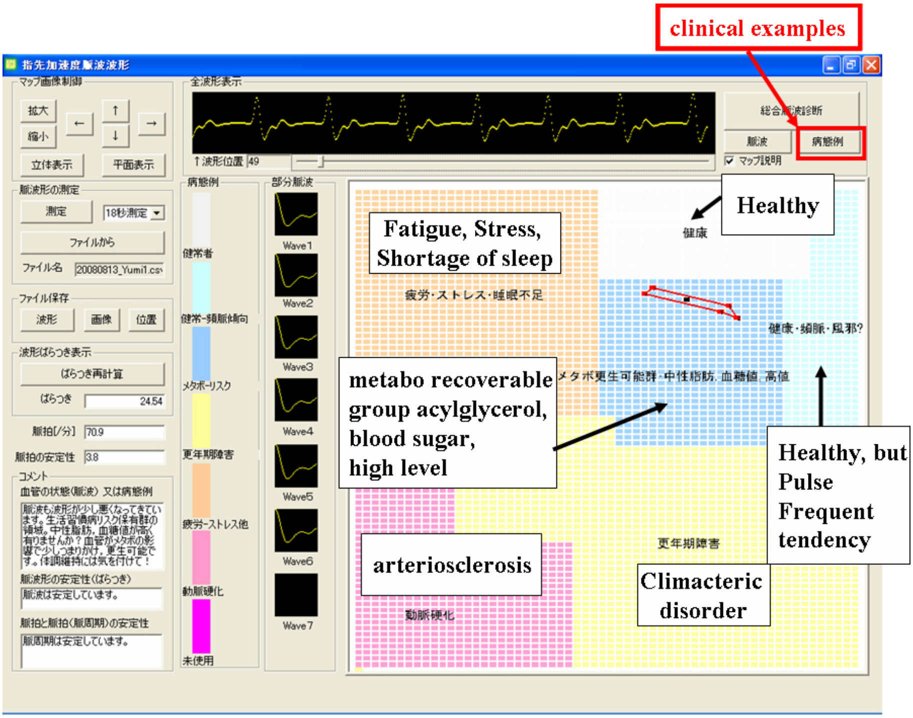
**Fig. 3.** Examples of a classification as healthy

predict the state of the blood vessel using Pulser_SOM. Pulser_SOM uses the plethysmogram sensor of U-Medica Inc. First, we explain what role each part of this software has. This software is developed based on the algorithm [5,6].

We show the example of the measurement in Fig. 2.

As shown in Fig. 2, the plethysmograms classify from "Group A" in the upper right to "Group G" in the lower left. The upper right section in each sub-figure of Fig. 3 shows an example of a healthy person. Dr. Maniwa, who is one of the authors, tried to classify the clinical examples of the plethysmogram [7]. Each region of the map in Fig. 4 shows a clinical example. As shown in Fig. 4, there are 6 clinically explained/relevant regions: "healthy", "healthy, but pulse frequent tendency", "metabo recoverable group: acylglycerol and/or blood sugar high level", "fatigue, stress, shortage of sleep", "climacteric disorder", "arteriosclerosis". Although the plethysmogram of the examinee in Fig. 2 is in the healthy region of "2", its clinical example in Fig. 4 is in the "metabo recoverable group".

As shown in Fig. 2, the position of the partial wave is shown by a point (The point can be expanded, so that the partial wave number becomes readable.) on the map. To the center of gravity of this point group corresponds the colors of A-G in the "legend", and a comment which corresponds to its color is shown in an editbox of "vascularity". In this research, the A-G colors are converted into numbers 1-7. The transition of the partial wave shown on the map is tied with the shortest distance as defined by TSP (Traveling Salesman Problem),



↑波形位置 49

18秒測定

ファイル名 20080813_Yumi1.cs\

ばらつき 24.54

脈拍[/分] 70.9

脈拍の安定性 3.8

コメント

血管の状態(脈波) 又は病態例

脈波も波形が少し悪くなってきています。生活習慣病リスク保有群の領域。中性脂肪，血糖値が高く有りませんか？血管がメタボの影響で少しつまりかけ，更生可能です。体調維持には気を付けて！

脈波形の安定性(ばらつき)

脈波は安定しています。

脈拍と脈拍(脈周期)の安定性

脈周期は安定しています。

**Fig. 4.** "The clinical example" of the examinee is shown by pushing button ③ in Fig. 2 which is the button of "the clinical examples". This examinee corresponds to the area of "metabo recoverable group" by the color of "the clinical examples" on the left of the figure. Also, the black mark surrounded by the line which is linked by five points of Wave1-Wave5 in the figure is the center of gravity (or the representative point ), where Wave6 is hidden among Wave1- Wave5. Putting the mark in the button ④ which is already shown in Fig. 2, the Japanese explanation of each color region on the map of "the clinical example" analysis mode of the examinee appears.

and its distance is shown by "looseness". Next "pulse" and "pulse stability" are also described by numbers. Using four items, and pushing the button ① of "Synthetic plethysmogram diagnosis" in Fig. 2, a new map is created (Details are omitted.). Figures 2, 3 and 4 are obtained from the Basic SOM model. For the Basic SOM model, the number of input data points is 1500, the number of neurons 60 80, the number of iterations 10,000, the learning coefficient 0.4, and the neighborhood radius 60, gaussian.

## 3 The Map Comparison among 1Map (Basic SOM), Torus and a Spherical SOM (Blossom)

When the map patterns of the pulse-waves are analyzed in BSOM, they are often collected to the upper part and the left edge on the map. Since the topological relation is kept more faithfully in Torus SOM and SSOM than in BSOM, the analyzed map positions are compared among them. As for Fig. 5, there are not so much differences among three SOMs. The Fig. 6 is the example where the map position is in the left edge of G area in BSOM. However, the map positions in Torus and SSOM are all in the F area. As a result of the detailed inspection, the above situations often occur in the upper left edge in the G area of BSOM.

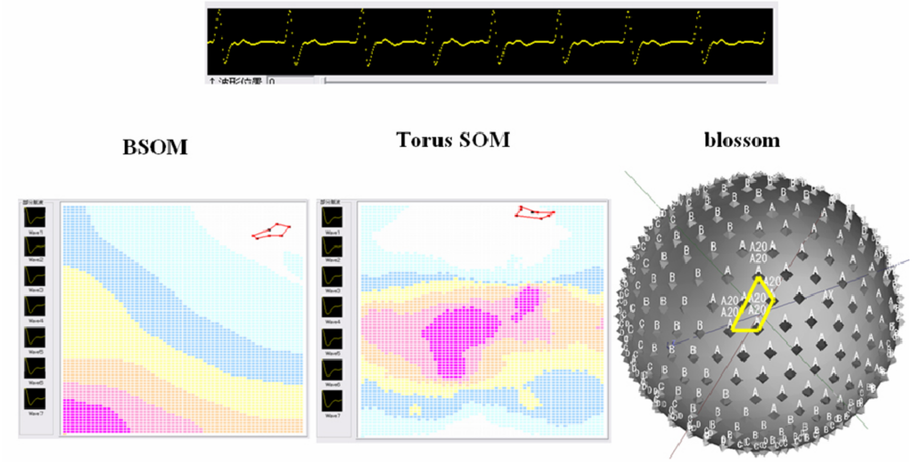


**Fig. 5.** The example which the distribution resembles well in BSOM, Torus, and SSOM

Here, we summarize the map comparison among 1map (Basic SOM), Torus, and a spherical SOM (blossom) [8].

1. For BSOM, the analysis result tends to gather to the upper part and the left edge.
2. The right edge of D in 1map connects with C from Torus/SSOM.
3. The left edge of G in 1map connects with F from Torus/SSOM.
4. Both Torus SOM and SSOM show the topologically similar distribution.
5. However, the shape of Torus SOM is complicated at present.
6. When putting best wave on the North Pole in SSOM, the worst wave is situated on the South Pole.
7. In this case, the wave type of A-C, and the part of D in Fig. 2 are situated on the Northern hemisphere and the part of D and E-G in Fig. 2 are positioned on the Southern Hemisphere.

Therefore, in the case of 1map, we cannot check the change of the acceleration plethysmogram with the lapse of time, because the analysis result gather to the upper part and the left edge. In the case of SSOM, we should rotate the map to check the whole map. As a result, Torus SOM is the best of all in this synthetic plethysmogram diagnosis.

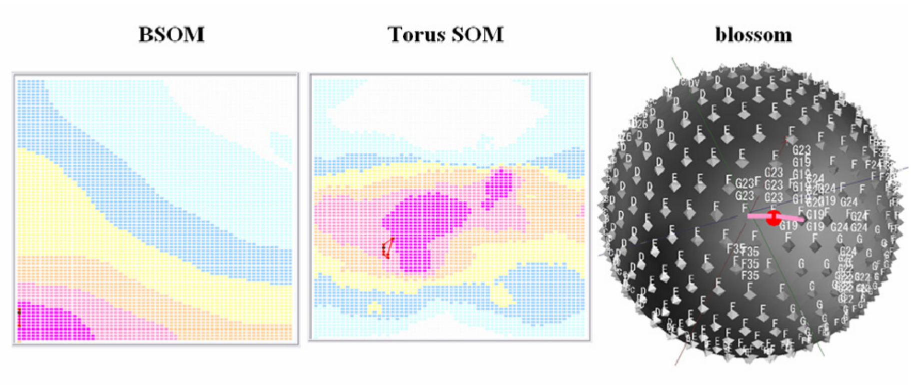


**Fig. 6.** In BSOM, when the analyzed positions are in the left end in the G area in Fig. 2, they are however, in the F area in Torus and SSOM. We should be careful because it often happens in the upper part in the left end in the G area.

## 4 Conclusion

We developed the diagnosis help software which shows the state of the blood vessel (vascularity) graphically using the Basic SOM model. In addition, we developed the classification of the clinical examples using the waveform of the plethysmogram. As a result, the examinee can assess the present health state quantitatively because the health state is given by a numerical value. We found that the clinical example classification provide suggestive evidence to medical doctors, researchers and non-experts. Moreover, we compared 1map (Basic SOM) with Torus and a Spherical SOM (blossom). We are convinced that Torus SOM is the best of all in this synthetic plethysmogram diagnosis.

## References

1. Sano, Y., Kataoka, Y., Ikuyama, T., Wada, M., Konno, H., Kawamura, K., Watanabe, T., Nishida, A., Koyamauchi, H.: The evaluation of the blood circulation and its application by the acceleration plethysmogram. The Journal of Science of Labour 61(3), 129–143 (1985) (in Japanese)
2. Kohonen, T.: Self-Organaizing Maps. Springer, Heidelberg (2001)
3. Tokutaka, H., Ohkita, M., Fujimura, K.: Self-Organizing Maps and its Application. Springer, Japan (2007) (in Japanese)
4. Takazawa, K., Fujita, M., Yabe, K., Sasaki, T., Kobayashi, T., Maeda, K.: Clinical usefulness of the second derivative of a plethysmogram. Cardiol 23(37), 207–217 (1993)
5. Noso, N., Fujimura, K., Ohkita, M.: Automatic Clustering on a Self-Organization Maps -Acceleration plethysmogram map-. In: Proceedings of Fuzzy System Symposium 2007, pp. 407–410 (2007) (in Japanese)

6. Noso, N.: Automatic Classification and Analysis of Acceleration Plethysmogram on a Self-Organization Maps. In: Proceedings of Acceleration Plethysmogram Complex Systems Workshop of Japan, p. 3 (2007) (in Japanese)
7. Tokutaka, H., et al.: Construction of Acceleration Plethysmogram Analysis System with SOM and the Visible Analysis of each Clinical Example. In: The 81st Japan Society for Occupational Health in Sapporo Convention Center, vol. E109, June 24-27 (2008) (in Japanese)
8. http://www.somj.com

# Decision of Class Borders on Spherical SOM and Its Visualization

Nobuo Matsuda[1], Heizo Tokutaka[2], and Matashige Oyabu[3]

[1] Oshima College of Maritime Technology, Information Science and Technology,
1091-1 Komatsu Suo-oshima-cho, Oshima-gun, Yamaguchi-ken, 742-2193, Japan
matsuda@oshima-k.ac.jp

[2] SOM JAPAN Co. Ltd.
680-0941 Tottori, Japan
tokutaka@somj.com

[3] Kanazawa Institute of Technology.
921-8501 Ishikawa, Japan
oyabu@neptune.kanazawa-it.ac.jp

**Abstract.** In this paper, we propose two methods for determining class borders. One approximates the borders on a Self-Organizing Map (SOM) from the nearest neighbor datasets of different classes. The other especially approximates the borders on a spherical SOM using the coordinate system of a polygon surface. Both methods decide the border on the SOM using the characteristics of the SOM which can map a high dimensional dataset onto a low dimensional map which is usually 2 or 3 dimensional. Using the iris dataset and the wine dataset, it is shown that both proposed methods allow the class borders to be successfully visualized in a comprehensible manner. The verification of the decision border, computed with one of the proposed methods, was performed with the spherical SOM and the dendrogram. The advantages of visualization and the improvement of the accuracy of a cluster analysis were successfully demonstrated using two benchmark databases.

## 1 Introduction

The Self-Organizing Map (SOM) is a powerful tool for exploring huge amounts of multi-dimensional data. The SOM by Kohonen [1] is a kind of neural network algorithm that projects high dimensional data onto a low dimensional space. In the traditional SOM algorithm, however, the "border effect" problem have been pointed out, and several spherical SOMs based on a geodesicdome [2] or a toroidal SOM have been proposed as a remedy. To show its potential effectiveness, the spherical SOM has been applied to clustering. For instance, Tokutaka *et al.* [3] proposed a highly accurate cluster analysis using the spherical SOM.

On the other hand, for the interpretation of information on the map, there is a new proposal [4], however the U-matrix [5] has been mainly used in the traditional SOM and the spherical SOM. In the U-matrix, the Euclidean distance between nodes is expressed by a gray level. Therefore, it is difficult to decipher the information of the class distribution or the border when the shading due to the U-matrix changes

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 802–811, 2009.

continuously. Tokutaka *et al.* [3] converted the shade of the U-matrix to the distance and obtained a dendrogram to perform a classification based on distances. They recommended discussing interactively the dendrogram and the graphical object on the polygon surface to eliminate misclassifications. In the discussion of their cluster analysis, they used boundaries that were artificially drawn. When discussing the dendrogram, however, it is crucial that the class boundary is decided accurately because the precision of the computed class boundaries controls the accuracy of the cluster analysis. Generally, it is difficult to depict the class borders of multi-dimensional data, however the projection capability of the SOM of multi-dimensional data will allow the class borders to be successfully visualized. Therefore, there exists a necessity for a method accurately drawing the decision borders on the SOM.

In this paper, to achieve the above goal, we propose new methods for determining the class decision borders. We especially present a new method using the coordinate system of the polygon to draw the decision border on the spherical SOM. We refer to here not only how to select candidates for determining the border but also how to examine the validness of the decision border. Then the proposed methods are applied to two benchmark datasets and the proposed ones are shown to be effective for visualizing of the decision borders of the classification analysis.

## 2 Methods and Procedures for Decision Borders of SOM Using Dataset of Boundary Neighborhood

### 2.1 Method for Decision Border Based on Input Dataset

Now, assume that there are two data points on the SOM. One of them is located at the center part of the class distribution, and the other is located near the class boundary. The latter will have an overwhelmingly larger influence than the former on the decision of the class border. Therefore, it is reasonable to approximate the class borders from the input data close to the boundary on the SOM.

Let us select a pair of feature vectors *X* and *Y* on the SOM as shown in Fig.1. They are of a different class (class A and class B) and are placed at the nearest boundary. Symbols of open triangle stand for class A and those of open square stand for class B, and symbols of open circle are candidate points as on the border. The feature vectors are assumed to be in m-dimensional and vector $X = (x_1, x_2, x_3, \ldots, x_m)$ belongs to class A and vector $Y = (y_1, y_2, y_3, \ldots, y_m)$belongs to class B.

Let us consider the case of determining point $D = (d_1, d_2, d_3, \ldots, d_m)$ of an arbitrary feature vector on the boundary from the feature vectors *X* and *Y*.

Assume that the two class distributions are almost equal, the point *D* on the boundary should be decided to satisfy the condition that the summation *S* of $k$ and $l$ is the minimal value, with $k$ is the square value of the distance between *D* and *X*, and $l$ is the square one of the distance between *D* and *Y*.

$$S = k + l = (x_1 - d_1)^2 + (x_2 - d_2)^2 + (x_3 - d_3)^2 + \cdots + (x_m - d_m)^2 + (y_1 - d_1)^2 + (y_2 - d_2)^2 + (y_3 - d_3)^2 + \cdots + (y_m - d_m)^2 \tag{1}$$

In order for *S* to be a minimal value, it is necessary to satisfy the simultaneous equations obtained by differentiating equation (1) with respect to $d_1$, $d_2$, $d_3$, …, $d_m$. Therefore, we get equation (2):

$$2(x_i - d_i) + 2(y_i - d_i) = 0 \quad (i = 1, m) \tag{2}$$

Finally, each components of point *D* is calculated by equation (3):

$$d_i = \frac{(x_i + y_i)}{2} \qquad (i = 1, m) \tag{3}$$

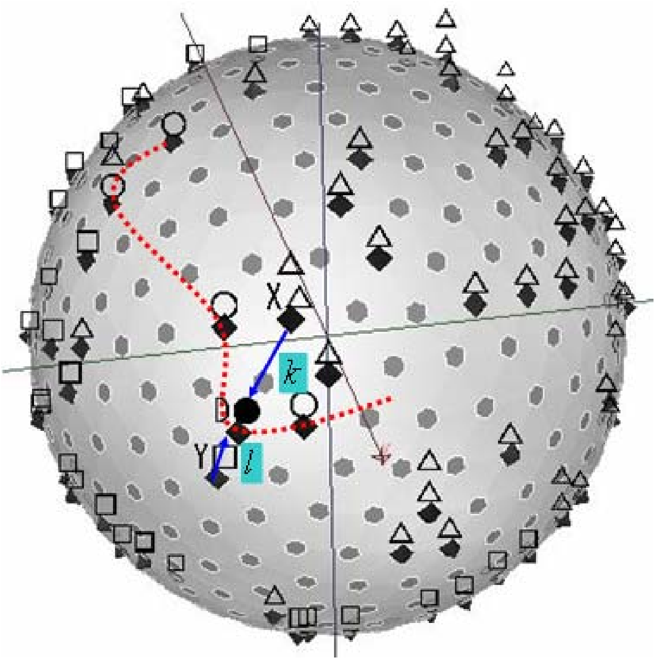


**Fig. 1.** Decision border and Feature vectors X and Y on the spherical SOM ( Open triangle nodes are class A and open square nodes are Class B. Open circle nodes are the calculated border and A close circle node is the point D. the dotted line is a part of the borderline).

## 2.2 The Procedure for Determining Decision Border Based on Input Dataset

When a probabilistic relation between the feature vector and the class distribution is uncertain, it is difficult to obtain the decision borders. Then it is necessary to make that the map contains the information of the distribution of the feature vector. The SOM is the easiest method to obtain such a map. After obtaining the SOM as shown in Fig.1, the following three steps are repeated for determining the decision borders.

Step 1

A number of candidates on the SOM (the traditional SOM or the spherical SOM) are selected from the data sets near the boundary.

Step 2

The distances between candidates are calculated and a pair of data points with the minimum distance is determined.

Step 3

Point D on the boundary is decided from the pair by equation (3).

After some points on the boundary are calculated by repeating step 1 through 3, a borderline is drawn. In Step 1, candidates in the class A and Class B are selected according to equation (4) among the datasets.

$$\exp(-(\mathbf{x}-\boldsymbol{\mu}_x)^2/R) \prec \varepsilon, \exp(-(\mathbf{y}-\boldsymbol{\mu}_y)^2/R) \prec \varepsilon. \tag{4}$$

where $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_y$ stand for vectorial reference points of each class, and $R$ stands for a parameter (The value within the range from 0.01 to 0.1 is usually used as a value of $R$ for the retrieval.). The vectorial reference points are chosen from the node located at the center part of the classes. $\varepsilon$ is threshold value. After some points on the boundary are calculated by repeating step 1 through 3, a borderline is drawn. In Step 1, candidates for the decision borders are usually selected among the boundary dataset. If necessary, nodes of SOM or other dataset can be chosen. They are selected on the basis of the distance between candidates.

### 2.3 Method for Decision Border Using the Coordinates on the Polygon Surface

Let us select a pair of nodes $P_k$ and $Q_k$ on the spherical surface are chosen in the boundaries neighborhood, as shown Fig.2. The nodes have 3-dimensional coordinate and node $P_k = (x_1, y_1, z_1)$ belongs to class A and node $Q_k = (x_2, y_2, z_2)$ belongs to class B. Assume that node $G_k = (x_0, y_0, z_{02})$ between $P_k$ and $Q_k$ is located at the node along the border on the polygon surface. Let us determine node $G_k$ from the coordinate system of the nodes $P_k$ and $Q_k$.

Firstly, a vector $\mathbf{a}$ is defined as a distance and direction between node $P_k$ and the center of the spherical object. Similarly, another vector $\mathbf{b}$ is defined between node $Q_k$ and the center of the spherical object. The following expressions (5) are derived from the mathematical and geometrical relations.

Where $h$ is the ratio of the magnitude of a vector $\mathbf{a}$ to one of a vector $\mathbf{b}$ from the center of spherical object at Glyph Analysis Setting of 1.0. The parameter of Glyph Analysis Setting stands for the transformation level on the sphere when blossom [6] is adapted to obtain the spherical SOM. Finally, each components of node $G_k$ is calculated by equation (6).

$$\begin{cases} \mathbf{C} = \mathbf{a} \times \mathbf{b} = \begin{vmatrix} i & j & k \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{vmatrix} = \alpha i + \beta j + \gamma k \\ \overrightarrow{OG_k} \cdot C = \alpha x_0 + \beta y_0 + \gamma z_0 = 0, \left|\overrightarrow{OG_k}\right| = r^2 = x_0^2 + y_0^2 + z_0^2 = 1 \\ \mathbf{a} \cdot \mathbf{b} = \cos\theta = x_1 x_2 + y_1 y_2 + z_1 z_2, \mathbf{a} \cdot \overrightarrow{OG_k} = \cos(h\theta) = x_0 x_1 + y_0 y_1 + z_0 z_1 \\ \mathbf{b} \cdot \overrightarrow{OG_k} = \cos(\theta - h\theta) = x_0 x_2 + y_0 y_2 + z_0 z \end{cases} \tag{5}$$

$$x_0 = \frac{(\beta z_1 - \gamma y_1) z_0 - \beta \cos(h\theta)}{(\alpha y_1 - \beta x_1)}, y_0 = \frac{(\gamma x_1 - \alpha z_1) z_0 + \alpha \cos(h\theta)}{(\alpha y_1 - \beta x_1)}, z_0 = \frac{-B \pm \sqrt{B^2 - AC}}{A} \tag{6}$$

Where $\alpha, \beta, \gamma, \theta, A, B, C$ are respectively

$$\begin{cases} \alpha = y_1 z_2 - z_1 y_2 \\ \beta = z_1 x_2 - x_1 z_2 \\ \gamma = x_1 y_2 - y_1 x_2 \end{cases}, \begin{cases} A = (\alpha y_1 - \beta x_1)^2 + (\beta z_1 - \gamma y_1)^2 + (\gamma x_1 - \alpha z_1)^2 \\ B = \cos(h\theta)\{\alpha \gamma x_1 + \beta \gamma y_1 - (\alpha^2 + \beta^2) z_1\} \\ C = (\alpha^2 + \beta^2)\cos^2(h\theta) - (\alpha y_1 - \beta x_1)^2 \end{cases}$$

$$\theta = \cos^{-1}(x_1 x_2 + y_1 y_2 + z_1 z_2) \tag{7}$$

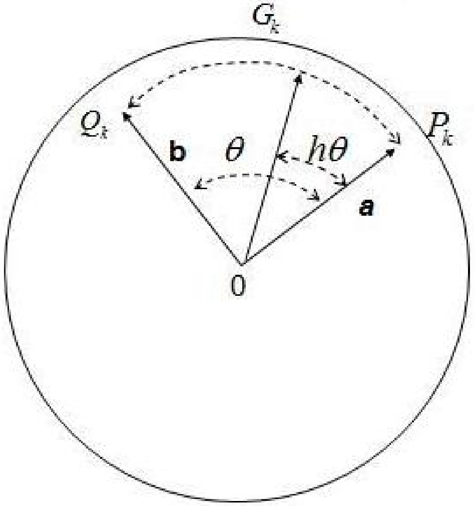


**Fig. 2.** Coordinate system of the polygon

### 2.4 The Procedure for Determining the Decision Border Based on the Coordinates

After selecting a pair of $P_k$ and $Q_k$ as candidates, the following three steps are repeated for determining the decision borders.

Step 1 Check and find out the coordinates of $P_k$ and $Q_k$ on the polygon surface.

Step 2 Obtain the coordinates of $P_k$ and $Q_k$ on the polygon surface using the Glyph Analysis Setting of 1.0.

Step 3 Measure two distances between the center of the sphere and $P_k$ and $Q_k$, and estimate coordinate of $G_k$ in accordance with the ratio of two distances using equation (6) through equation (7).

### 2.5 Verification of Decision Border

The verification of the decision border determined with equation (3) is performed with the dendrogram of the input dataset. The process of verification is as follows.

1) The nodes with different distribution ratios are calculated in the route of a pair of candidates.
2) The nodes calculated by step 1), the node D on the boundary and original datasets are learned using spherical SOM.
3) The dendrogram is constructed and is examined.

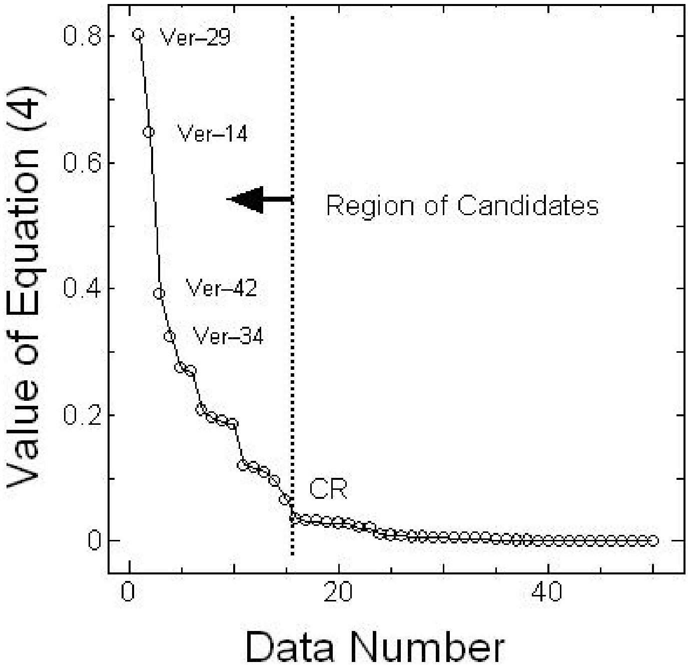


**Fig. 3.** Selection of candidates ($R = 0.01$, $\boldsymbol{\mu}_x = (0.4820, 0.3311, 0.6063, 0.5829)$, $\varepsilon = 0.0092$)

# 3 Database for Analysis and Results

## 3.1 Dataset for Analysis

The two kinds of benchmark dataset for analysis are described: the iris dataset of Fisher [7] is a well known benchmark data and consists of three classes (setosa (50), virginica (50), vergicolor (50) ). The data is in 4-dimensional and the attributes are sepal length and its width, and are petal length and its width. The other benchmark dataset used the wine dataset [7]. The data consists of three class wines (wine 1 (59), wine 2 (71), wine 3 (48)).

## 3.2 Result Based on Feature Vectors

For the iris data, the points from b0 to b4 on the spherical SOM were calculated by repeating steps 1 through 3 for the combination of the versicolor datasets and the virginica datasets. The calculated points form the borderline near the four data sets, versicolor19, 23, 24 and virginica20, were misclassified in the cluster analysis using the spherical SOM [3]. Fig.3 shows the candidates were selected based on the equation (4) when node b0 on the boundary was determined.

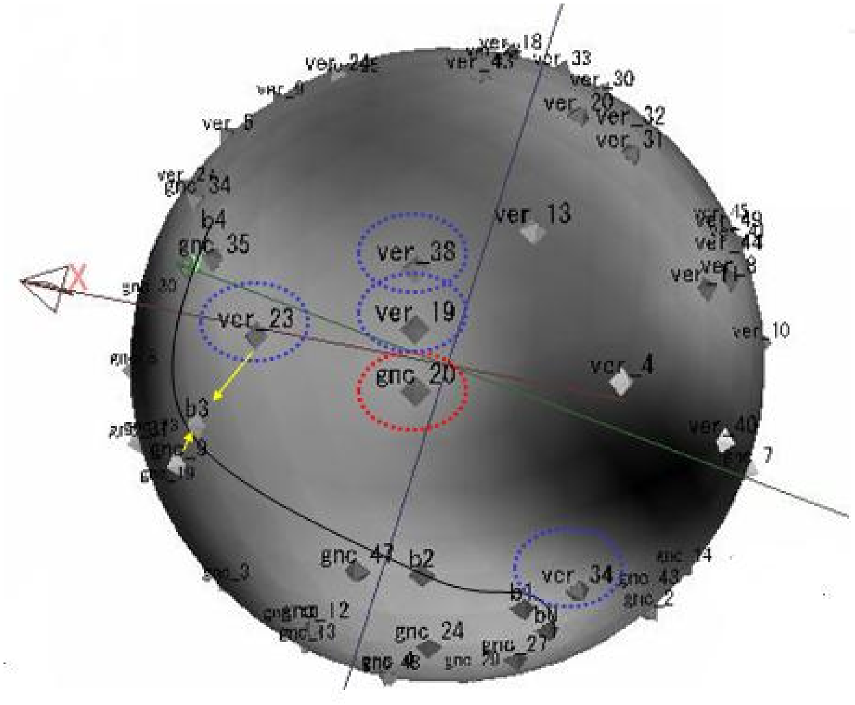


**Fig. 4.** A borderline pictured in the vicinity of gnc_20

We adopted the candidates among the nodes that were contained within the left side region from the point CR. In this figure the slope of the value calculated by equation (4) significantly changed at this point CR. Fig.4 shows the results of the projection of the calculated values onto the spherical SOM. In this figure, gnc_20 was misclassified as a vergicolor class by the cluster analysis using the spherical SOM. We can see from this figure that the borderline can be successfully expressed in the ambiguous shade region of the U-matrix.

### 3.3 Result for the Decision Border Based on the Coordinate System

Fig.5 shows the decision borders determined with the coordinate system on the polygon surface. The dotted decision border with nodes from b0T to b4T was determined with equation (6-7) and the solid decision border with the nodes from b0 to b4 was also determined with equation (3). When comparing the decision borders determined with two methods, the values of three of the five nodes, b0T, b3T and b4T, on the decision border are a very good match with the corresponding values of b0, b3 and b4 but the other ones of the nodes, b1T and b2T, do not match with the correspond values of b1 and b2.

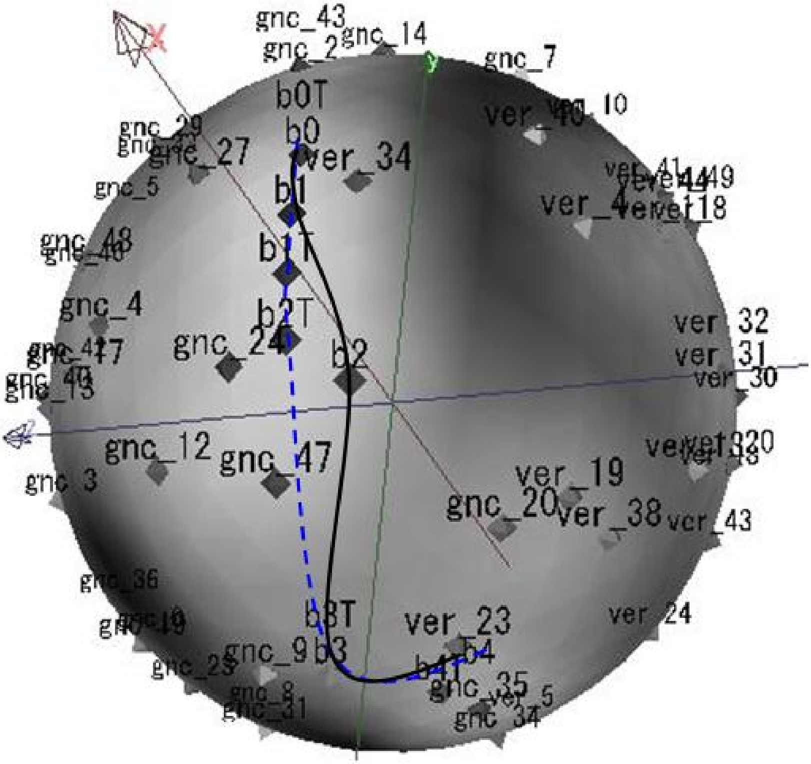


**Fig. 5.** The decision borders were obtained with two proposed methods

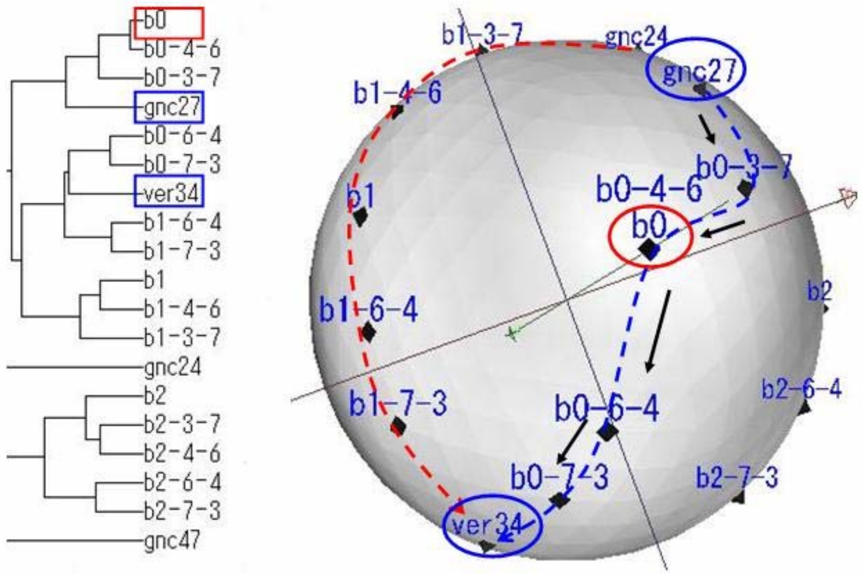


**Fig. 6.** Spherical SOM (right) and its dendrogram (center). They were made up from five border nodes (b0, b1, b2, b3, b4), twenty quantile-nodes and seven candidates.

### 3.4 Verification of the Decision Border by Dendrogram

The verification of the decision border was performed with b0 to b4 of the iris database using the dendrogram. Now, assume that a node divides internally the distance between a pair of candidates in the ratio of p to q. If nodes with the ratio of p>q belong to one branch of the dendrogram, nodes with the ratio p<q should belong to another branch because a dividing node with the ratio p = q define as a border node by equation (3). Then the nodes with four kinds of dividing ratios (4:6, 3:7, 6:4, and 7:3) were determined on the route between a pair of candidates(ver34,gnc27), and the dendrogram was obtained under the condition of the multivariate analysis of the Group Average at the Glyph Analysis Setting of 1.0.

Fig.6 shows the result of the verification of the decision border determined using the proposed method with the dendrogram. A red circle (b0) and a red box (b0) indicate the border and two blue circles (ver34, gnc27) and two blue boxes (ver34, gnc27) indicate candidates in the figure. Label b0_p_q in the figure describes a node with the internally dividing ratio p:q. We can confirm that b0 is the decision border between gnc27 and ver34 from this dendrograms.

### 3.5 Results of Application to Cluster Analysis

The application to cluster analysis of the proposed method was performed for the iris and wine database. When the cluster analysis for the iris dataset was performed by using the spherical SOM, the maximum result was given 97% in accuracy: Five datasets, gnc_7 in the virginica class, ver_19, 23, 34, 38 in the vergicolor class, were misclassified by analyzing only the dendrogram. When the cluster analysis is, however, performed by the decision borders using the proposed methods, the four datasets, ver_19, 23, 34, 38, were accurately classified as shown in Fig.4, and gnc_7 was also accurately classified. Consequently only gnc_20 was misclassified and yielded 99% accuracy. The analysis for the wine database was also performed with the proposed method and the perfect classification was attained.

## 4 Conclusions

We have proposed two methods which approximates the decision borders on a SOM from the nearest neighbor datasets of the different classes. The advantages and capabilities of the both proposed methods were successfully demonstrated using two benchmark datasets. Using the iris dataset of Fisher and the wine dataset, it was shown that both proposed methods allow the class borders to be successfully visualized and to be comprehensible. The verification of the decision border determined using one of the proposed methods was performed with the spherical SOM and the dendrogram. Moreover, when the proposed methods were applied to the cluster analysis using the spherical SOM, it was also shown that the accuracy of the cluster analysis is further improved: the accuracy was up 99% from 97% for the iris database and the cluster analysis for the wine database attained a perfect classification.

## References

1. Kohonen, T.: Self-Organizing Maps. Springer Series in Information Sciences, vol. 3.0 (2001)
2. Nakatsuka, D., Oyabu, M.: Usefulness of Spherical SOM for Clustering. In: Proceedings 19th Fuzzy System Symposium, pp. 67–70 (2003)
3. Tokutaka, H., Fujimura, K., Ohkita, M.: Cluster Analysis using Spherical SOM. Journal of Biomedical Fuzzy Systems Association 8(1), 29–39 (2006) (in Japanese)
4. Ultsch, A., Mörchen, F.: ESOM-Maps: tools for clustering, visualization, classification with Emergent SOM, Depart. Of Computer Science University of Marburg, Research Report 46 (2005)
5. Ultsch, A., Guimaraes, G., Korus, D., Li, H.: Knowledge extraction from artificial neural networks and applications. In: Proceedings of TAT/ WTC 1993, pp. 194–203. Springer, Heidelberg (1993)
6. http://www.somj.com/
7. http://www.ics.uci.edu/~mlearn/databases /

# Quantifying the Path Preservation of SOM-Based Information Landscapes

Michael Bui and Masahiro Takatsuka

ViSLAB
School of Information Technologies, University of Sydney, NSW 2006, Australia
mbui@vislab.usyd.edu.au, masa@vislab.usyd.edu.au

**Abstract.** Spatialization methods create visualizations that allow users to analyze high-dimensional data in an intuitive manner and facilitates the extraction of meaningful information. Just as geographic maps are simplified representations of geographic spaces, these visualizations are essentially maps of abstract data spaces that are created through dimensionality reduction.

Recently, we proposed to use the spherical Geodesic SOM for creating an information landscape that represents an abstract data space and can capture a manifold's global structure. Path finding was then applied to approximate the geodesic path in the feature space and some promising preliminary results were obtained. Based on these results, we propose a novel approach for measuring the preservation of geodesics by analyzing the paths on the information landscapes. The effectiveness of the measure is then evaluated through various data sets.

## 1 Introduction

Information visualization has been broadly defined as "a computer-aided process that aims to reveal insights into an abstract phenomenon by transforming abstract data into visual-spatial forms" [1]. Tory et al. categorize spatializations into two groups [2] based on the representation of the data:

- Points: where the data is represented by a set points
- Information landscapes: where a surface attempts to fit the data set

They highlight the lack of empirical evidence to the benefits of information landscapes that are often stated in literature, such as how they call upon our innate abilities of pattern recognition and spatial reasoning [3]. Nevertheless, users expect that distances between points on spatializations to correspond to high-dimensional similarities [4] and abundant in literature are a variety of techniques for measuring the correlation between the two. However, according to our knowledge, all of these measures are for point spatializations and no measure exists for information landscapes.

In this paper, we present a new technique that fills this gap in the literature and is applicable to information landscapes generated by the Self-Organizing

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 812–819, 2009.

Map algorithm. The rest of the paper will be structured as follows: in the next section, we review the work related to that presented in this paper, including the Self-Organizing Map algorithm and a number of popular topology preservation measures. Afterwards, the main contribution of our paper is presented. Namely, a technique for measuring how well an information landscape preserves geodesics. We then present the results of experimenting with various data sets to evaluate the performance of our technique, and then provide some conclusions and discussions on future avenues of research.

## 2 Related Work

### 2.1 Self-Organizing Map

The SOM is a popular artificial neural network that can be used to visualize data. This comes from SOM's following characteristics: (1) being able to preserve topology in the data, (2) the capability to generalize data (since the weight vectors tend to approximate the probability function of the input vectors), (3) performing multidimensional scaling and (4) unsupervised clustering.

The conventional SOM consists of a two-dimensional grid of $N$ neurons arranged in a rectangular or hexagonal manner. Each neuron is associated with a weight vector $w_j$ of the dimensionality as the input vectors $x_i$. They are typically initialized to random values before training the SOM with the data set. During the training process, an input vector is presented to the SOM and its best matching unit (BMU) is computed. The BMU $c$ is the neuron with the weight vector closest to the input vector according to a distance function. The input vector is then mapped to the neuron's sublist of points. The values of the weight vectors belonging to the BMU $c$ and the neurons within its neighbourhood are then adjusted to be closer to the input vector's values. Typical implementations use a Gaussian neighbourhood that decreases over time.

This process is applied to the all input vectors and the SOM is usually trained over a large number of epochs until convergence is reached. Additional visualization techniques can then be applied to further aid the user in understanding the structure of the high-dimensional data through visual inspection. The U-Matrix [5] is the most commonly used technique, which visualizes the local distances (U-heights) in high-dimensional space in the form of a landscape or colour.

### 2.2 Topology Preservation Measures

SOMs produce topology preserving mappings of high-dimensional spaces. While this capability enables it to produce a visual representation of a high-dimensional space, it would be scientifically comforting to have a measure that indicates how accurate or truthful the visual appearance is to the original data space. In order to evaluate the quality of the mappings, several measures have been proposed to quantify the degree of topology preservation.

**Topographic Product.** The topographic product [6] is one of the oldest and widely used topology preservation measures. It is used to measure the differences between the dimensionality of the network (output space) and the dimensionality of the manifold (input space). This involves comparing the relationship between each pair of neurons in both spaces. Villmann et al. [7] have shown that the topographic product only works for nearly linear data sets. For the topographic product to provide accurate results for nonlinear spaces, Revuelta et al. [8] proposed to use geodesic distance on the manifold in the computation of the topographic product. This can be approximated by calculating shortest path on the induced Delaunay triangulation graph, which will be described later on.

**Topographic Error.** The topographic error is a simple measure that quantifies the continuity of the mapping. Given a data sample $x$ from the data set $X$, find the nearest weight vector $w_i$ and second nearest weight vector $w_j$. If the corresponding neurons $i$ and $j$ are adjacent on the neural lattice, then the mapping is locally continuous, otherwise there is a local topographic error. The topographic error $\varepsilon_t$ is obtained by taking the average of the local topographic errors for all data samples $x$. However, the results for smaller maps can be unreliable as the first and second BMU are more likely to be adjacent.

**Topographic Function.** Villman et. al [7] introduced a topology preservation measure called the topographic function based on the induced Delaunay triangulation graph $G$ of the weight vectors. They define weight vectors $w_i$ and $w_j$ as being adjacent on the manifold $V$ if their receptive fields are adjacent. The adjacency of the receptive fields can be determined by computing the connectivity matrix $C$ of the induced Delaunay triangulation graph $G$:

1. Given a data sample $x$, find its first BMU $i$ and second BMU $j$
2. Create a synaptic link between neurons $i$ and $j$, i.e. set $C_{ij} = 1$
3. Go back to step 1 and repeat for all data samples

If the number of neurons/weight vectors is “dense” enough on the manifold $V$, then the graph $G$ represents a perfect topology preserving mapping of $V$ that also preserves the paths on $V$. The topographic function can be used on non-linear spaces since the induced Delaunay triangulation graph contains information on the shape of the input space. It is capable of measuring the neighbourhood preservation from the input space $V$ to the output space $A$ and vice versa. Furthermore, it can indicate the range of the largest fold if the $dim(V) > dim(A)$ or the range of the topology preservation violations if $dim(V) < dim(A)$.

**Topological Correlation.** The topological correlation method [9] is another measure that relies on the induced Delaunay triangulation graph. It measures the linear correlation between the topological distances (minimum path length) in the network graph and induced Delaunay triangulation graph. Using the index allows users to identify a perfect mapping by quantifying how well the network graph matches the induced Delaunay triangulation graph. Furthermore, the maximum number of neurons to use can be identified.

# 3 A Path Preservation Measure for the SOM

## 3.1 Approximating Geodesic Paths Using Information Landscapes

A survey on topology preservation measures reveals a large number of techniques that have successfully been used for quantifying the relationship between distances between points on spatializations and the distance in the original feature space. The results of these measures would allow users to determine whether or not the visualization faithfully captures the topological relationship before performing any exploratory data analysis. However, all of these techniques have thus far been applied to point spatializations.

Earlier work on using the SOM for geospatial analysis [10] demonstrated that distances in the original feature space could be perceived as geodesic distances on the synthetic U-Matrix landscape. This provided the motivation in our recent work where a path finding algorithm was applied on the U-Matrix for trajectory analysis [11]. Our preliminary results showed some promising signs that indicated that geodesic paths could be approximated through this approach. Therefore, information landscapes could potentially better capture the topological relationships within the data.

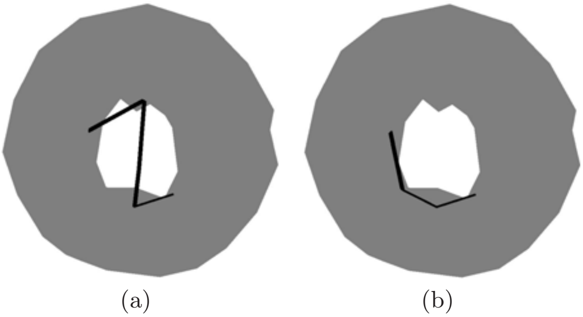


**Fig. 1.** An example of finding the geodesic path between two points on a torus by (a) calculating the shortest path on the Geodesic SOM and (b) incorporating U-heights to find the shortest path

Figure 1 demonstrates an example. Here, a Geodesic SOM has been trained with points sampled from the surface of a torus. The figure depicts the reconstructed surface based on the induced Delaunay triangulation graph. If one were to calculate the shortest path in the feature space between two neurons by calculating the shortest path between the two neurons on the neural lattice, then it is possible for the path to ignore the topology of the data space (Figure 1 (a)). On the other hand, using the U-heights as the cost function to calculate the shortest path [11] generates a path that better respects the topology (Figure 1 (b)).

### 3.2 Landscape Distance Correlation Measure

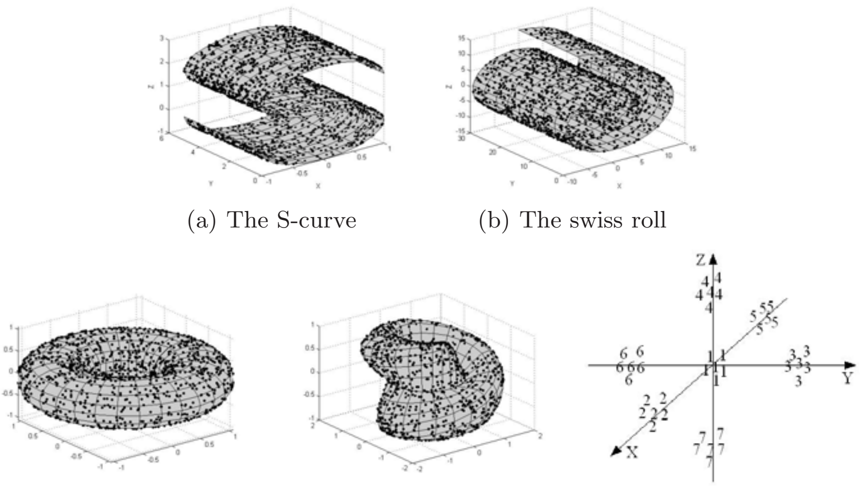
With this in mind, it would be of interest to derive a quantitative measure that can evaluate the approximated paths using information landscapes. Our approach to calculating this, called the landscape distance correlation measure, relies on computing the induced Delaunay triangulation graph $G$ of the weight vectors after training the SOM. Then for a neuron $i$, we attempt to measure the local correlation measure between it, and every other neuron $j$. If no path exists between neuron $i$ and neuron $j$ $(j \neq i)$ on graph $G$, then the local correlation measure has a value of 0. However, if the path exists, then we calculate the geodesic path from $i$ to $j$ on $G$ and sum the distances in the feature space between consecutive neurons. This distance $d^G$ represents the real geodesic distance on the manifold. We then approximate the geodesic path between $i$ and $j$ using the U-Matrix landscape similar to the approach in [11]. However, floodplain analysis isn't used as we are interested in seeing if the paths do cross cluster boundaries. The SOM is thus treated as a weighted graph where the cost to go from neuron $a$ to $b$ is the latter's U-height $uh(b)$. Given a goal neuron $i$, the distance transformation algorithm, which is similar to Dijkstra's algorithm, is applied to compute a distance map. A path from $j$ to $i$ can then be traced by following the steepest descent from $j$ on the distance map. Our approximated geodesic distance $d^L$ is computed by summing the distances in the feature space between consecutive neurons on the path. The local correlation value is $\frac{d^L(i,j)}{d^G(i,j)}$. This is done for all neurons and the equation for our measure is:

$$L_C = \frac{1}{(N-1)!} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{d^L(i,j)}{d^G(i,j)} \tag{1}$$

where $L_C = 0$ indicates no correlation, $L_C = 1$ indicates perfect correlation and $L_C > 1$ means the $d^L$ is on average $L_C$ times larger than $d^G$.

## 4 Experiments

In order to evaluate the results of our measure, we experimented with various 3D data sets. This allows us to visually inspect the results (estimated paths). Four of these data sets consists of 2000 points that were randomly sampled on the S-curve, swiss roll, torus and Klein bottle manifold with a uniform distribution. The last data set consists of 7 well separated clusters (called the 7 balls data set for reference), each of which contain 500 uniformly distributed points. The standard deviation in each dimension is 1 and the centre of the clusters are (0, 0, 0), (10, 0, 0), (0, 10, 0), (0, 0, 10), (-10, 0, 0), (0,-10, 0) and (0, 0, - 10). This data set has previously been used to illustrate how the Geodesic SOM can reveal more information about how the data is related to each other. The four data sets are shown Figures 2 (a), (b), (c) and (d) respectively. We also experimented with n-dimensional data (the ionosphere data set) obtained from the UCI machine learning repository [12].



(a) The S-curve

(b) The swiss roll

(c) The torus

(d) The Klein bottle

(e) The 7 balls synthetic data set and how some of the points are arranged.

**Fig. 2.** Manifolds and sampled points used for training the Geodesic SOM

For all data sets, a three-frequency geodesic dome is used (91 neurons), the initial update radius is 9, initial learning rate is 1 and 1000 epochs are used for training using the batch update method. The initial values of the weight vectors were initialized using PCA.

Table 1 contains the results of our experiments. The $L_C$ measure indicates that the geodesic distance $d^L$ calculated using the U-Matrix landscape can be quite close to the geodesic distance $d^G$ calculated using the induced Delaunay triangulation graph. This occurs as the approximated paths tend to follow the induced Delaunay triangulation graph (refer to Figure 1 above). For comparison, we have calculated another distance correlation value $D_C$ that uses the

**Table 1.** The computed landscape topological correlation values for each data set. $D_C$ is a distance correlation value that uses the shortest paths calculated using the positions of the neurons on the lattice.

| Data set | $L_C$ | $D_C$ |
|---|---|---|
| S-curve | 0.910351 | 1.330177 |
| Swiss roll | 1.072374 | 1.562333 |
| Torus | 1.134007 | 1.636815 |
| Klein | 1.162603 | 1.707200 |
| 7 balls | 0.142587 | 0.320250 |
| Ionosphere | 0.778884 | 1.334701 |

same equation above. In this case though, the value of $d^G(i, j)$ is calculated by finding the shortest path between $i$ and $j$ on the network graph and taking its corresponding distance in the feature space as the value. The results indicate that using the U-Matrix landscape will generally provide better approximations than using the network graph method.

The $L_C$ measure for the 7 balls data set returns a very low correlation value. This occurs as based on the lattice connectivity, paths exist between the 7 well-separated clusters even though the induced Delaunay triangulation graph can consists of 6 subgraphs that approximates the number of clusters in the data space. In these situations, it may be of interest to measure the correlation of the intra-cluster distances. In other words the local correlation for a pair of neurons will only be measured if they belong to the same cluster according to the graph $G$. Using this method returned a value of 1.139066, which indicates a good correlation. The $D_C$ value on the other hand is much worse compared to before (2.55833).

## 5 Conclusions and Future Work

It has previously been demonstrated that path finding techniques can be used with the SOM to approximate geodesics on a manifold. This can be achieved by path finding techniques on the U-Matrix landscape. In order to evaluate the accuracy of these paths, a proper measure is required.

In this paper, we have proposed a new measure that quantifies the correlation between the approximated geodesic distance after performing path finding on an information landscape generated by the SOM and the geodesic distance on the induced Delaunay triangulation. The latter acts as a sort of ground truth, which may not be readily be available for evaluation in many applications. Our experiments confirm the effectiveness of this measure.

One of the disadvantages of this approach is that calculating induced Delaunay triangulation requires a suitable amount of neurons with respect to the number of data samples and this value is difficult to determine. Although more accurate

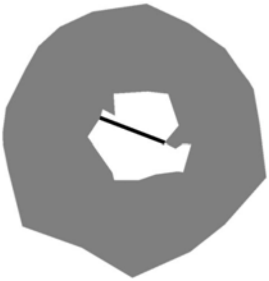


**Fig. 3.** An example of a path approximated using the landscape method depicted on the induced Delaunary triangulation graph of the torus that ignores the topology of the data

approximated geodesic distances can be obtained with larger SOMs, applying this measure would not be feasible as the induced Delaunay triangulation graph would consist of many dead neurons and disconnected subgraphs.

Another disadvantage is that our technique only compares the distances and not the paths themselves to see if it follows the structure of the manifold (Figure 3). A possible avenue for further research would be determining a more complex measure that solves the latter problem.

## References

1. Chen, C.: Information visualization. Information Visualization 1(1), 1–4 (2002)
2. Tory, M., Sprague, D., Wu, F., So, W.Y., Munzner, T.: Spatialization design: Comparing points and landscapes. IEEE Transactions on Visualization and Computer Graphics 13(6), 1262–1269 (2007)
3. Spence, R.: Information Visualization. Addison Wesley, Reading (2000)
4. Montello, D.R., Fabrikant, S.I., Ruocco, M., Middleton, R.S.: Testing the first law of cognitive geography on point-display spatializations. In: Kuhn, W., Worboys, M.F., Timpf, S. (eds.) COSIT 2003. LNCS, vol. 2825, pp. 316–331. Springer, Heidelberg (2003)
5. Ultsch, A., Siemon, H.P.: Exploratory data analysis: Using kohonen networks on transputers. Technical Report 329, University of Dortmund (1989)
6. Bauer, H.U., Pawelzik, K.: Quantifying the neighborhood preservation of self-organizing feature maps. IEEE Transactions on Neural Networks 3(4), 570–579 (1992)
7. Villmann, T., Der, R., Herrmann, M., Martinetz, T.M.: Topology preservation in self-organizing feature maps: exact definition and measurement. IEEE Transactions on Neural Networks 8(2), 256–266 (1997), doi:10.1109/72.557663
8. Revuelta, F.F., Chamizo, J.M.G., Rodríguez, J.G., Sáez, A.H.: Geodesic topographic product: An improvement to measure topology preservation of self-organizing neural networks. In: Lemaître, C., Reyes, C.A., González, J.A. (eds.) IBERAMIA 2004. LNCS (LNAI), vol. 3315, pp. 841–850. Springer, Heidelberg (2004)
9. Doherty, K., Adams, R., Davey, N.: Topological correlation. In: ESANN 2006: Proceedings of the 14th European Symposium on Artificial Neural Networks, April 2006, pp. 125–130 (2006)
10. Takatsuka, M.: An application of the self-organizing map and interactive 3-d visualization to geospatial data. In: Proceedings of the 6th International Conference on GeoComputation (September 2001)
11. Bui, M., Takatsuka, M.: Path finding on a spherical SOM using the distance transform and floodplain analysis. In: Proceedings of the Workshop on Self-Organizing Maps, WSOM 2007 (2007)
12. Asuncion, A., Newman, D.: UCI machine learning repository (2007)

# Self-Organizing Neural Grove and Its Parallel and Distributed Performance

Hirotaka Inoue

Department of Electrical Engineering and Information Science,
Kure National College of Technology,
2-2-11 Agaminami, Kure-shi, Hiroshima, 737-8506 Japan
hiro@kure-nct.ac.jp

**Abstract.** In this paper, we present the improving capability of accuracy and the parallel efficiency of self-organizing neural groves (SONGs) for classification on a MIMD parallel computer. Self-generating neural networks (SGNNs) are originally proposed on adopting to classification or clustering by automatically constructing self-generating neural tree (SGNT) from given training data. The SONG is composed of plural SGNTs each of which is independently generated by shuffling the order of the given training data, and the output of the SONG is voted all outputs of the SGNTs. We allocate each of SGNTs to each of processors in the MIMD parallel computer. Experimental results show that the more the number of processors increases, the more the classification accuracy increases for all problems.

## 1 Introduction

Neural networks have been widely used in the field of intelligent information processing such as classification, clustering, prediction, and recognition. Generally, these neural networks have to be decided the network structure and some parameters by human experts. It is quite tricky to choose the right network structure suitable for a particular application at hand. Concerning the design of the network structure, the following must be decided, (i) the number of the network layers, (ii) the number of the neurons of each layer, (iii) the weights on connection between consequent layers. During learning iterations, the weights on connections of the given networks are updated so as to converge to target value conserving the initially decided static network structure. Consequently, obtaining the right structure of each network is the most important factor in learning and also the most difficult problem in the design of neural networks.

In order to avoid these tricky and difficult situations, Self-generating neural networks (SGNNs) are focused an attention because of their simplicity on networks design [1]. SGNNs are some kinds of extension of the self-organizing maps (SOMs) of Kohonen [2] and utilize the competitive learning algorithm which is implemented as self-generating neural tree (SGNT).

The SGNT algorithm is proposed in [3] to generate a neural tree automatically from training data directly. In our previous study concerning the performance

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 820–827, 2009.

analysis of the SGNT algorithm [4], we showed that the main characteristic of this SGNT algorithm was its high speed convergence in computation time but it was always not best algorithm in its accuracy comparing with the existing other feed-forward neural networks such as the back-propagation (BP).

In order to improve the generalization capability of SGNNs, we proposed ensemble self-generating neural networks (ESGNNs) for classification [5]. ESGNNs apply ensemble averaging [6] to SGNNs and fully utilize the high speed convergence characteristics of the SGNT algorithm. Although ESGNNs are improved the accuracy by using various SGNTs, the computation time and the memory capacity are increased in proportion to increase the number of SGNTs. Therefore, we proposed a novel pruning method for the structure of the ESGNNs to reduce the computation time and the memory capacity and we called this model as self-organizing neural grove (SONG) [7].

Ensemble learning has been studied many AI and neural network researchers. Breiman proposed bagging predictors to improve the accuracy of CART [8] and investigated bagging performance on CART and other methods for classification and regression problems in [9]. Since ensemble learning is a variance-reduction technique, it is well known that ensemble learning tends to work well for methods with high variance such as neural networks and tree-based methods.

In this paper, we present the improving capability of accuracy and the parallel efficiency of the SONG for classification on a MIMD parallel computer. We apply to three problems in the UCI repository [10] which are given as benchmark.

## 2 Self-Organizing Neural Grove

In this section, we describe how to prune redundant leaves in the SONG. First, we mention the on-line pruning method in learning of SGNT. Second, we show the optimization method in constructing the SONG.

### 2.1 On-Line Pruning of Self-Generating Neural Tree

SGNT is based on SOM and implemented as a competitive learning. The SGNT can be constructed directly from the given training data without any intervening human effort. The SGNT algorithm is defined as a tree construction problem of how to construct a tree structure from the given data that consist of multiple attributes under the condition that the final leaves correspond to the given data.

Before we describe the SGNT algorithm, we denote some notations.

- input data vector: $\boldsymbol{e}_i \in \mathbb{R}^m$.
- root, leaf, and node in the SGNT: $n_j$.
- weight vector of $n_j$: $\boldsymbol{w}_j \in \mathbb{R}^m$.
- the number of the leaves in $n_j$: $c_j$.
- distance measure: $d(\boldsymbol{e}_i, \boldsymbol{w}_j)$.
- winner leaf for $\boldsymbol{e}_i$ in the SGNT: $n_{win}$.

```
Input:
  A set of training examples E = {e_i},
 i = 1, ... , N.
  A distance measure d(e_i,w_j).
Program Code:
  copy(n_1,e_1);
  for (i = 2, j = 2; i <= N; i++) {
    n_win = choose(e_i, n_1);
    if (leaf(n_win)) {
      copy(n_j, w_win);
      connect(n_j, n_win);
      j++;
    }
    copy(n_j, e_i);
    connect(n_j, n_win);
    j++;
    prune(n_win);
  }
Output:
  Constructed SGNT by E.
```

**Fig. 1.** SGNT algorithm

The SGNT algorithm is a hierarchical clustering algorithm. The pseudo C code of the SGNT algorithm is given in Fig. 1. In Fig. 1, several sub procedures are used. Table 1 shows the sub procedures of the SGNT algorithm and their specifications.

In order to decide the winner leaf $n_{win}$ in the sub procedure `choose(e_i,n_1)`, the competitive learning is used. If an $n_j$ includes the $n_{win}$ as its descendant in the SGNT, the weight $w_{jk}$ $(k = 1, 2, \ldots, m)$ of the $n_j$ is updated as follows:

$$w_{jk} \leftarrow w_{jk} + \frac{1}{c_j} \cdot (e_{ik} - w_{jk}), \quad 1 \leq k \leq m. \tag{1}$$

After all training data are inserted into the SGNT as the leaves, the leaves have each class label as the outputs and the weights of each node are the averages

**Table 1.** Sub procedures of the SGNT algorithm

| Sub procedure | Specification |
|---|---|
| $copy(n_j, \boldsymbol{e}_i/\boldsymbol{w}_{win})$ | Create $n_j$, copy $\boldsymbol{e}_i/\boldsymbol{w}_{win}$ as $\boldsymbol{w}_j$ in $n_j$. |
| $choose(\boldsymbol{e}_i, n_1)$ | Decide $n_{win}$ for $\boldsymbol{e}_i$. |
| $leaf(n_{win})$ | Check $n_{win}$ whether $n_{win}$ is a leaf or not. |
| $connect(n_j, n_{win})$ | Connect $n_j$ as a child leaf of $n_{win}$. |
| $prune(n_{win})$ | Prune leaves if the leaves have the same class. |

of the corresponding weights of all its leaves. The whole network of the SGNT reflects the given feature space by its topology. For more details concerning how to construct and perform the SGNT, see [3]. Note, to optimize the structure of the SGNT effectively, we remove the threshold value of the original SGNT algorithm in [3] to control the number of leaves based on the distance because of the trade-off between the memory capacity and the classification accuracy. In order to avoid the above problem, we introduce a new pruning method in the sub procedure `prune(n_win)`. We use the class label to prune leaves. For leaves connected to the $n_{win}$, if those leaves have the same class label, then the parent node of those leaves is given the class label and those leaves are pruned.

In the next sub-section, we describe how to optimize the structure of the SGNT in the SONG to improve the classification accuracy.

## 2.2 Optimization of the SONG

The SGNT has the capability of high speed processing. However, the accuracy of the SGNT is inferior to the conventional approaches, such as nearest neighbor, because the SGNT has no guarantee to reach the nearest leaf for unknown data. Hence, we construct an MCS by taking the majority of plural SGNT's outputs to improve the accuracy.

Although the accuracy of the SONG is comparable to the accuracy of conventional approaches, the computational cost increases in proportion to increase in the number of SGNTs in the SONG. In particular, the huge memory requirement prevents the use of the SONG for large datasets even with latest computers. In order to improve the classification accuracy, we propose an optimization method of the SONG for classification. This method has two parts, the merge phase and the evaluation phase. The merge phase is performed as a pruning algorithm to reduce dense leaves (Figure 2). This phase uses the class information and a threshold value $\alpha$ to decide which subtree's leaves to prune or not. For leaves that have the same parent node, if the proportion of the most common class is greater than or equal to the threshold value $\alpha$, then these leaves are pruned and the parent node is given the most common class. The optimum threshold values $\alpha$ of the given problems are different from each other. The evaluation

```
begin   initialize $j$ = the height of the SGNT
  do for each subtree's leaves in the height $j$
    if the ratio of the most class $\geq \alpha$,
    then merge all leaves to parent node
    if all subtrees are traversed in the height $j$,
    then $j \leftarrow j - 1$
  until $j = 0$
end.
```

**Fig. 2.** The merge phase

```
begin initialize $\alpha = 0.5$
  do for each $\alpha$
    evaluate the merge phase with 10-fold CV
    if the best classification accuracy is obtained,
    then record the $\alpha$ as the optimal value
    $\alpha \leftarrow \alpha + 0.05$
  until $\alpha = 1$
end.
```

**Fig. 3.** The evaluation phase

phase is performed to choose the best threshold value by introducing 10-fold cross validation (Figure 3).

## 3 Parallel and Distributed Processing

Because each SGNT of the SONG can train and test independently, the SONG has a possibility of the parallel computation at the training process and the testing process. Hence, we allocate each of SGNTs to each of processors on the MIMD computer. The procedure of the parallelization of the SONG is presented as follows:

**Step1:** In a master processor, read the training set $D$ and the test set $T$ in the disk.
**Step2:** In the master processor, broadcast $D$ and $T$ for all $K-1$ slave processors.
**Step3:** In all processors, generate the SGNT from $D$, then test the SGNT using $T$, and compute the $\boldsymbol{o}_k$ independently.
**Step4:** In all processors, each output $\boldsymbol{o}_k$ for $T$ is collected in the master processor by all to one communication.
**Step5:** In the master processor, compute $\boldsymbol{o}$ by voting and write to the disk.

Because the number of the communications between the master processor and each slave processor is only two times (Step2 and Step4), the parallel efficiency is approximately expected the linear speedup. In our case, all computations are performed on the Intel Paragon (Paragon XP/S15). This is a distributed memory multicomputer, and the architecture is multiple instruction multiple data (MIMD). The Paragon we use has 296 processors. Each processor is Intel i860XP (50MHz). The network topology of the Paragon is adopted the two-dimensional mesh.

## 4 Experimental Results

We allocate each of SGNTs to each of processors on the Paragon, and compute 100 trials for each single/ensemble model. The number of processors (SGNTs) $K$

for the ensemble averaging is changed from 1 to 201 (1,3,5,7,9,15,25,51,101,151 and 201), and the threshold value $\alpha$ is 1 for each SONG. In order to reduce the redundant execution, we repeated 100 trials from Step3 to Step5 in prior section continuously.

In order to investigate the parallel performance of the SONG, we select three typical classification problems, which are given as benchmark problems in UCI repository [10]. Next, we describe the brief explanation of these problems.

**breast-cancer-wisconsin:** This problem is a binary classification task for classify a tumor as either benign or malignant based on cell descriptions gathered by a microscopic examination. Input attributes are:

- the clump thickness,
- the uniformity of cell size,
- cell shape,
- the amount of magical adhesion,
- the frequency of bare nuclei, etc.

This problem has 9 attributes, 699 examples. Each attribute consists of continuous real value.

**ionosphere:** This problem is a binary classification task for a radar as either good or bad based on the complex electromagnetic signals. The targets were free electrons in the ionosphere. “Good” radar returns are those showing evidence of some type of structure in the ionosphere. “Bad” returns are those that do not; their signals pass through the ionosphere. This problem has 34 attributes, 351 examples. Each attribute consists of continuous real value.

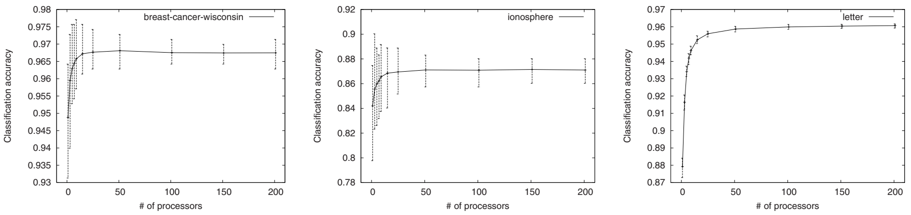
**letter-recognition:** The objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts), which were then scaled to fit into a range of integer values from 0 through 15.

In this paper, we use below defined classification accuracy.

$$\text{classification accuracy} = \frac{\text{number of correct}}{\text{number of test data}} \,. \qquad (2)$$

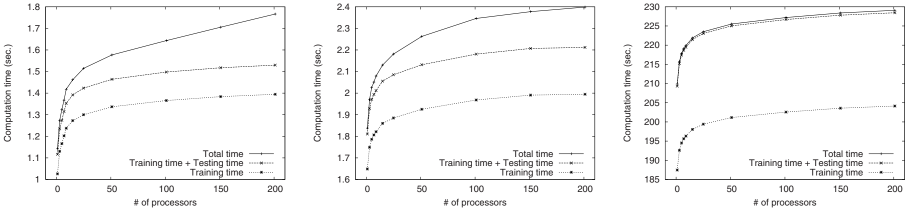
We evaluate the classification accuracy using 10-fold cross-validation [11] for above problems.

Figure 4(a), (b), and (c) show the influence of the number of processors on classification accuracy for breast-cancer-wisconsin, ionosphere and letter-recognition problems respectively. Classification accuracies are improved by computing the ensemble averaging of various SGNTs for all problems. Here, each classification accuracies shows the average of 100 trials and its error-bar. It is shown that the classification accuracies are improved by computing the ensemble averaging of various SGNTs for all problems. Especially, the minimum classification



(a) (b) (c)

**Fig. 4.** Influence of the number of processors on classification accuracy for (a) breast-cancer-wisconsin, (b) ionosphere, and (c) letter-recognition



(a) (b) (c)

**Fig. 5.** Relation between the number of processors and execution time (seconds) for (a) breast-cancer-wisconsin, (b) ionosphere, and (c) letter-recognition

accuracies are largely improved for all problems. The improvement ability is obtained from small $K$ most effectively. The classification accuracy of larger than 51 SGNTs is convergence for all problems.

Figure 5(a),(b), and (c) show the relation between the number of processors and the execution times for breast-cancer-wisconsin, ionosphere, and letter-recognition problems respectively. The execution times are gradually saturated as the number of processors increase. As the scale of the dataset grows, the proportion of the communication time, i.e. the difference of the total time and the training time + the testing time, for the total time is decrease. This means that this method have an approximately linear speedup for large-scale datasets.

Consequently, a parallel distributed computing using SONG can obtain more higher classification accuracy than the single SGNT by allocating each of SGNTs to each of processors, go on maintaining the high speed processing property of the single SGNT.

## 5 Conclusions

In this paper, we presented the parallel distributed computing with SONG to obtain more effective implementation for classification on the MIMD parallel computer. From the experimental results the following conclusions can be drawn:

- Parallel distributed computing with SONG can improve the classification accuracy using various SGNTs which are allocated processors on the MIMD computer.
- Parallel distributed computing with SONG can perform a task with the high parallel efficiency by allocating each of SGNTs to each of processors on the MIMD computer.

In the future work, we will study an incremental learning of SONG for large-scale data mining.

## Acknowledgements

The authors would like to thank the Information Processing Center in Okayama University of Science for using the Paragon.

## References

1. Wen, W.X., Pang, V., Jennings, A.: Self-generating vs. self-organizing, what's different? In: Simpson, P.K. (ed.) Neural Networks Theory, Technology, and Applications. IEEE Technology Update Series, pp. 210–214. IEEE Technical Activities Board, Piscataway (1996)
2. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (1995)
3. Wen, W.X., Jennings, A., Liu, H.: Learning a neural tree. In: The International Joint Conference on Neural Networks, November 3-6, vol. 2, pp. 751–756. Beijing, China (1992)
4. Inoue, H., Narihisa, H.: Efficiency of Self-Generating Neural Networks Applied to Pattern Recognition. Int. J. of Mathematical and Computer Modelling 38(11–13), 1225–1232 (2003)
5. Inoue, H., Narihisa, H.: Improving generalization ability of self-generating neural networks through ensemble averaging. In: Terano, T., Chen, A.L.P. (eds.) PAKDD 2000. LNCS (LNAI), vol. 1805, pp. 177–180. Springer, Heidelberg (2000)
6. Haykin, S.: Neural Networks: A comprehensive foundation, ch. 7, 2nd edn. Prentice-Hall, Upper Saddle River (1999)
7. Inoue, H.: Self-organizing neural grove: Efficient multiple classifier system with pruned self-generating neural trees. In: Proc. the 18th International Conference on Artificial Neural Networks, Part I, Prague, Czech Rep., September 3–6, pp. 613–622 (2008)
8. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth, California (1984)
9. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
10. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
11. Stone, M.: Cross-validation: A review. Math. Operationsforsch. Statist., Ser. Statistics 9(1), 127–139 (1978)

# The Finding of Weak-Ties by Applying Spherical SOM and Association Rules

Takaichi Ito[1] and Tetsuya Onoda[2]

[1] Graduate School of Media and Governance, Keio University
kiichi@sfc.keio.ac.jp
[2] School of Information-Oriented Management, Sanno University
ond@ond-lab.net

**Abstract.** The finding of an evident relation in the same category is trivial. But the finding of an unexpected relation between different categories is interesting. It is called “Weak-ties” of the network analysis with the unexpected relation between different categories.

In this research, we apply Spherical Self-Organizing Map (SOM) and Association Rules to find "Weak-ties". Spherical SOM is applied to explain macro structure, and Association Rules are applied to find micro short-cuts. By applying those two methods, we are able to find relations that are distant on macro structure but strong in micro relevance.

We applied this method to the cognition data of comic books (Manga) and found "Weak-ties" of two and three of comic artists that are different in genre, generation and sex.

**Keywords:** Data Mining, Self-Organizing Maps, Association Rules, Small World.

## 1 Introduction

In the digitalizing society, the technology that determines findings from the data or data mining technology is in demand with digital data being easily available in large quantity.

However, there are some problems in the methods of data mining. For instance, "Association Rules"[1] is not capable to express a macro structure still it outputs a large amount of objectively explainable micro fact, and, on the contrary, "Self-Organizing Maps"[2] tells a macro structure as it does not show micro facts. This phenomenon leads us to the vision that the gap between structure and relevance is the key to findings.

It is trivial when there is no gap between the structure and the relevance. "Diapers and Beer" is a widely known phenomenon on "Association Rules" and it was once taken as a surprising finding because they were linked in spite of their completely different categories. As in this case, items that are linked in a same category or not linked in different categories are not Findings. Findings must be unexpected and unpredictable.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 828–835, 2009.

It is inappropriate to tell that something is a finding without the definition and the understanding of "findings". "Weak-ties"[3], the technical term of network analysis, indicates a short-cut connection between different categories. Applying "Weak-ties" is one of directions to search a finding.

# 2 Methodology

## 2.1 Small World and Spherical SOM (Self-Organizing Maps)

Watts's "Small world" model [4] was precedent in the field of Complex Networks study.

"Small world" means the situation that the distance between each node is small with a few edges and the clusterization of a large quantity of nodes.

Through the study, Watts and Strogatz discovered WS model that explains "Small World" in a very simple way. WS model is that circularly arranged nodes with edges on the circle turned into "Small World" by randomly switching edges. (Fig.1) The Intermediate condition between regular graph and random graph is "Small World". Within a "Small World", edges that connect distant nodes are "Weak-ties".

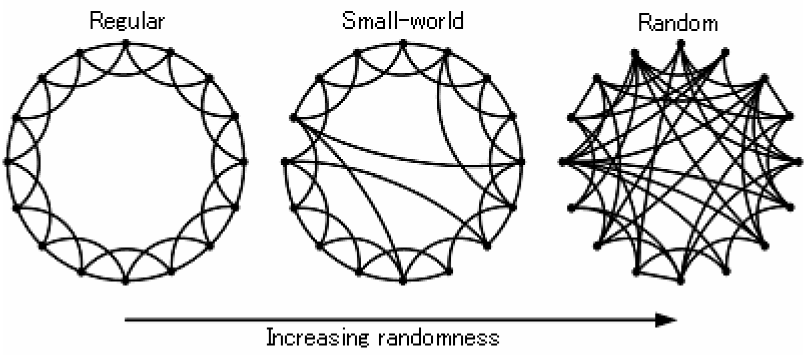


**Fig. 1.** Watts and Strogatz model (WS model)

Now, it is the question about how it is possible to construct the circle of WS model with an actual data set.

One of answers is to apply the self-organizing, or self-organized sequence. The circle is a collective of overlapping clusters and it is synonymous to the aspect of the topological geometry of SOM.

However, an ordinary SOM plotted on a frat-screen is not capable to describe the circle of WS model. In the first place, it is not circular, and a SOM plotted on a flat-screen has the aspect that the map is distorted on its margin and sometimes its opposite margins are approximate. A data-set figured a circle with chain like sequence is difficult to plot on a flat-screen.

To avoid distorted mapping at the margin of a flat-screen, spherical SOM [5] is efficient. By SOM plotted with spherical shape, the distortion is avoidable because of the absence of map margins on a spherical map. In addition, Spherical description of

WS model is the extension of the circle of WS model. WS model is described as a circle due to its convenience for describing on the flat-screen, and spherical map is not structurally different to it.

This idea leads us to the conclusion that "Weak-Ties" led from WS model is also able to be found through the examination of the relevance between nodes on the spherical SOM.

Above discussion is about generating spherical SOM from WS model. At the same time, it leads the discussion of plotting the data-set with small world structure on a spherical surface.

Therefore, unlike a regular network, the result of spherical SOM indicates the fact that not only approximate nodes but also distant nodes are associated. Besides, with clustering induced by self-organizing, spherical SOM is different from random network but it is rather the middle between regular and random, or "Small World" in terms of WS model.

### 2.2 Combining Spherical SOM and Association Rules

The followings are our suggestion in this paper. The finding of weak-ties is done by applying the combination of spherical self-organizing and association rules, in the same way as the occurrence of "Weak-ties" in WS model.

Taking the distance on the spherical SOM and the strength of relevance in association map as each axes of a Cartesian coordinate, we obtain the next four quadrants (Fig.2). The phenomenon plotted in the 4th quadrant, suggesting approximate distant in the self-organizing map and strong co-occurrence relevance, is trivial. It is also same in the 2nd quadrant as it suggests remote distance in the self-organizing map and thin co-occurrence relevance.

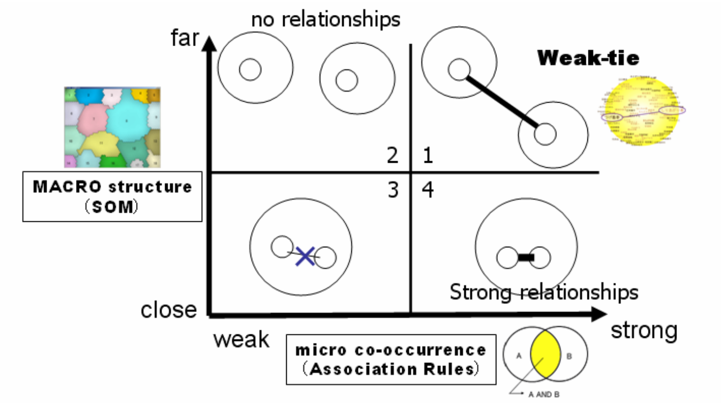


**Fig. 2.** Combination of “Sphere SOM” and “Association Rules”

The phenomenon plotted in the 1st quadrant suggests distant relevance in the self-organizing map and strong co-occurrence, and it is very "Weak-ties". In other words, it is the finding of the gap between macro structure of SOM and micro association relevance in association rules.

Here, we apply Euclidean distance to measure macro distance on the SOM. The radius of the spherical SOM is 1.0, then the maximum distance between certain two nodes is 2.0 that is the diameter of the sphere. In the case of micro co-occurrent relevance, "Jaccard" is applied. Given class A and B, Jaccard is represented by (1).

$$|A\cap B|/(|A|+|B|-|A\cap B|) \qquad \therefore |A\cap B|/|A\cup B| \tag{1}$$

When it comes to the co-occurence relevance of three, the displacement of distance to area(S) by using Heron's (2) is applied. In terms of co-occurrence probability, the use of co-occurrence product is efficient.

$$S=\sqrt{s(s-a)(s-b)(s-c)} \qquad \left(s=\frac{1}{2}(a+b+c)\right) \tag{2}$$

### 2.3 Explanation of Weak-Ties

The definition of "Weak-ties" is a short cut that connects distant nodes. For the explanation of it, let related items be explaining variables. Assume that the "Weak-tie" is found between item A and item B, then the difference between items approximate to A and items approximate to B are illustrated to clarify that item A and item B are in different categories.

Here, the term "approximate" implies the 4th quadrant that indicates approximate distance on the map and strong co-occurrence relevance.

Therefore, the distance on the map and the strength of co-occurrence relevance must be compared. Consequently, the Evaluation of the fraction of distances between the certain item C and item A, and item C and item B must be done. The result of it is the index of distance ratio. (3)(4)

$$\text{distance ratio} = \text{Dist (A,C)} / \{ \text{Dist (A,C)} + \text{Dist (B,C)} \} \tag{3}$$

$$\text{distance ratio} = \text{Dist (B,C)} / \{ \text{Dist (A,C)} + \text{Dist (B,C)} \} \tag{4}$$

In terms of co-occurrence probability, let the comparative indicator be "co-occurrence ratio". When Jac is the function to evaluate co-occurrence probability, co-occurrence ratio is able to be illustrated by (5).

$$\text{co-occurrence ratio} = \text{Jac (A,C)} / \text{Jac (B,C)} \tag{5}$$

If the value of it is greater than 1, then C has relatively stronger co-occurrence relevance with A, in reverse, C is more strongly connected with B compared to A when the value is less than 1.

By using "distance ratio" and "co-occurrence ratio", the extraction of approximate associations to item A and item B in the 4th quadrant is done. Through understanding of found approximate associations, the aspects of each category are explainable.

The next step is the extraction of items that are common to A and B. For the extraction of appropriate items, the co-occurrence product that multiplies each co-occurrence is applied. Co-occurrence product is capable to describe greater value when the co-occurrences on item A and item B are both have high frequency. With applying this method, it is figured that C has aspects of both A and B.

## 3 Case Examples

### 3.1 Applied Data Set

The data set used in this research is the response data to the questionnaire of iMap.gr.jp regarding comics ("manga").[6] iMap.gr.jp is the web site which made the large-scale survey on cultural recognition in 2000.

The questionnaire was done by checking and choosing appropriate items that respondents remembered or had the impression on.

After data cleaning, the data includes responses on the recognition to 420 comic artists by 3164 respondents. As a preprocessing, the frequency of response was logarithmically-transformed and equally divided into five before clustering them. This preprocessing reduced the difference of sizes of nodes plotted on the SOM by reducing the effect of frequency in the process of clustering.

The spherical SOM without this clustering process did not draw effectual arrangement of nodes but nodes were arranged according to their frequency. We identified responses that indicate at least one of comic artists who are involved in the cluster as 1, and other responses that do not fulfill the condition as 0.

On the basis of that, we regarded responses of whole respondents as feature vectors (3164 dimensions) and drew the spherical SOM.
Figure 3 shows the result.

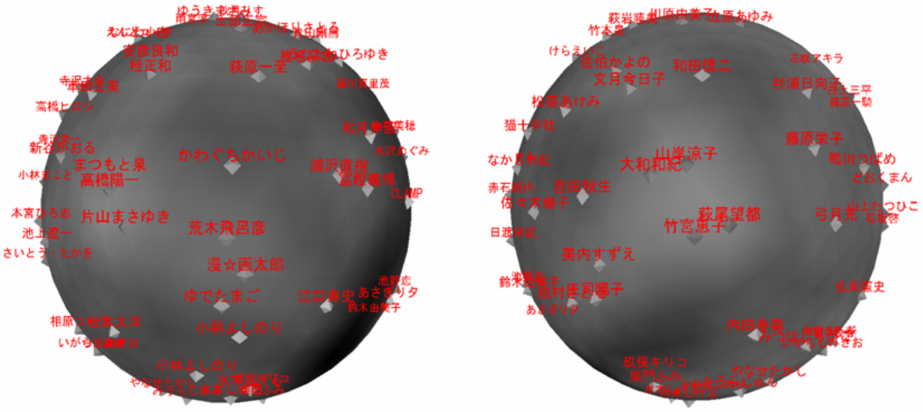


**Fig. 3.** The result of the spherical SOM

### 3.2 Weak-Ties between Two

The list of couples of nodes which have distant interval and strong co-occurrence relevance among the nodes on the map is as follows.

**Table 1.** The result between two

| Manga-artist A | Manga-artist B | Distance | Co-occurrence |
|---|---|---|---|
| Shiriagari Kotobuki | Tsuge Yoshiharu | 1.957 | 0.300 |
| Yuzuki Hikaru | Eguchi Hisashi | 1.672 | 0.307 |
| Yudetamago | Yuzuki Hikaru | 1.517 | 0.300 |

Now by the consideration of relevance of distance ratio and co-occurrence ratio between "Shiriagari Kotobuki" and "Tsuge Yoshiharu" who are plotted completely opposite each other on the SOM, we found following insights. (Fig.4.)

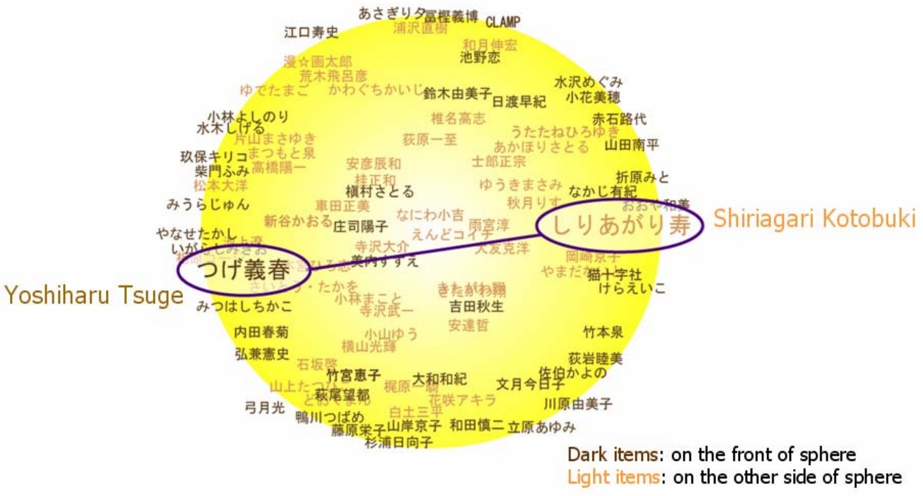


**Fig. 4.** Geometry on Spherical SOM of “ShiriagariKotobuki” and” Yoshiharu Tsuge”

Shiriagari Kotobuki’s masterpiece is "Yaji & Kita in midnight" and he is plotted close to comic artists whose works were popular with matured women, mainly with working women born around in 1970.

By contrast, Tsuge Yoshiharu, whose masterpiece is "Neji-shiki", is popular amongst Baby-boomers, who were born between late 1940s and early 1950s.

Here, Shiriagari’s fans and Tsuge’s fans are completely different in terms of generation and sex, still both comic artists are connected by the method we took in this research.

Looking at comic artists who have strong relevance common to them with using co-occurrence product relevance, we found that their common denominators are commercial comic "Garo", surrealistic atmosphere on their works, and the fact that they have been often cited by same comic critic.

### 3.3 Weak-Ties among Three

Table 2 shows the top 8 list of the extraction of the combinations that have areas of over 1.0 and co-occurrence product of over 0.15.

Here, we made the brief discussion on the relevance among "Urasawa Naoki", "Sasaki Tomoko" and "Aihara Koji" that has the highest renown index in total. These three comic artists construct a huge triangle on the sphere.

**Table 2.** The result among three

| Manga-Artist A | Manga-Artist B | Manga-Artist C | Support | Relation | A - B | B - C | C - A | Index | Value |
|---|---|---|---|---|---|---|---|---|---|
| Urasawa Naoki | Sasaki Noriko | Aihara Koji | 23.2% | Distance | 1.478 | 1.663 | 1.473 | Area | 1.014 |
| | | | | co-occurrence | 0.293 | 0.211 | 0.275 | Product | 0.017 |
| Yukimasami | YudeTamago | YumitsukiHikaru | 21.5% | Distance | 1.342 | 1.517 | 1.898 | Area | 1.010 |
| | | | | co-occurrence | 0.309 | 0.300 | 0.236 | Product | 0.022 |
| YumitsukiHikaru | Eguchi Hisashi | Kurumada Masami | 21.0% | Distance | 1.672 | 1.362 | 1.639 | Area | 1.027 |
| | | | | co-occurrence | 0.307 | 0.280 | 0.246 | Product | 0.021 |
| Yukimasami | YumitsukiHikaru | Eguchi Hisashi | 20.9% | Distance | 1.898 | 1.672 | 1.406 | Area | 1.138 |
| | | | | co-occurrence | 0.236 | 0.307 | 0.232 | Product | 0.017 |
| Eguchi Hisashi | Shintani Kaoru | Wada Shinji | 18.1% | Distance | 1.317 | 1.850 | 1.816 | Area | 1.126 |
| | | | | co-occurrence | 0.259 | 0.283 | 0.226 | Product | 0.017 |
| Okazaki Kyoko | Eguchi Hisashi | Utida Shungiku | 17.5% | Distance | 1.731 | 1.321 | 1.885 | Area | 1.104 |
| | | | | co-occurrence | 0.201 | 0.279 | 0.291 | Product | 0.016 |
| Shintani Kaoru | Hiwatari Saki | Wada Shinji | 16.1% | Distance | 1.894 | 1.274 | 1.850 | Area | 1.121 |
| | | | | co-occurrence | 0.206 | 0.291 | 0.283 | Product | 0.017 |
| Nagai Go | Shintani Kaoru | Tachihara Ayumi | 14.2% | Distance | 1.317 | 1.730 | 1.849 | Area | 1.091 |
| | | | | co-occurrence | 0.268 | 0.250 | 0.255 | Product | 0.017 |

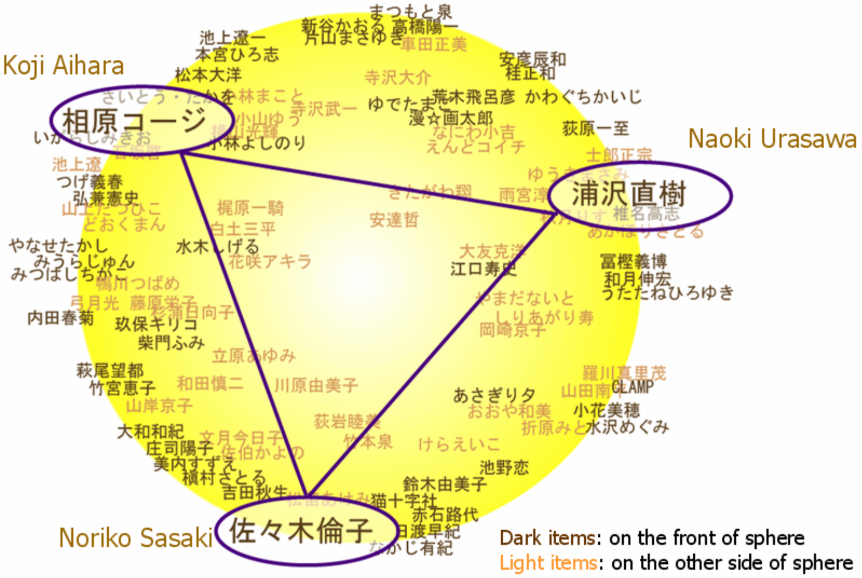


**Fig. 5.** Geometry on Spherical SOM of "Urasawa Naoki", "Sasaki Tomoko" and "Aihara Koji"

Urasawa's masterpiece is "Yawara", the story of genius judo heroine. Urasawa excels at story comic for young male people. Inoue Takehiko, the author of "SLAM DUNK" is close to him on the SOM.

Sasaki, whose masterpiece is "Veterinarian school story", is surrounded by the artists of girl's comic whose works take importance on story and less on love romance.

Aihara excels at gag strip for rather matured people, and his masterpiece is "How to draw comics easily".

The "Weak-tie" found among these artists, is the fact that their works appeared serially on "Big Comic Spirits", the profitable magazine published by Shogakukan. Main readers of this magazine are the young male business people and college students. Noriko Sasaki was strategically headhunted by the editorial department.

## 4 Summary

In our research, we have observed the weak-ties among three or two artists. As a summary, by applying this approach, it is able to find weak-ties among artists, even if they belong to different genres, generations and seemingly unrelated when observed from the macro structure.

Factors of the occurrence of "Weak-ties" are the brand of the magazine, criticism and marketing strategy.

As an overall summary, "Weak-ties" over genres in general is discovered by this approach.

In addition, the approach of finding 'Weak-ties' can be applied not only comics but many other fields.

## References

[1] Agrawal, R., Imielinski, T.: Mining association rules between sets of items in large databases. In: Proceedings of ACM SIGMOD International Conference on Management of Data (1993)

[2] Kohonen, T.: Self Organizeing Maps. Springer, Heidelberg (1995)

[3] Granovetter, M.: The Strength of Weak Ties. American Journal of Sociology 78(6), 1360–1380 (1973)

[4] Watts, D., Strogatz, S.: Collective dynamics of small world networks. Nature 393, 440–442 (1998)

[5] Nakatsuka, D., Oyabu, M.: Application of Spherical SOM in Clustering. In: Proceedings of Workshop on Self-Organizing Maps (WSOM 2003), pp. 203–207 (2003)

[6] Ito, T., Onoda, T.: Finding "weak-ties" via structuring the Manga phenomenon based on reader'scognition of Manga artists. Manga. Studies 13, 26–64 (2008)

# Analysis of Robustness of Pareto Learning SOM to Variances of Input Vectors

Hiroshi Dozono[1] and Masanori Nakakuni[2]

[1] Faculty of Science and Engineering, Saga University,
1-Honjyo Saga 840-8502 Japan
hiro@dna.ec.saga-u.ac.jp
[2] Information Technology Center, Fukuoka University,
8-19-1 Nanakuma, Jounan-ku, Fukuoka 814-0180 Japan
nak@fukuoka-u.ac.jp

**Abstract.** We have proposed Supervised Pareto learning Self Organizing Maps (SP-SOM) which based on the concept of Pareto optimality for the integration of multiple vectors and applied SP-SOM to biometric authentication system which used multiple behavior characteristics as feature vectors. As a variant of SP-SOM, we have also proposed Supervised Full Pareto learning SOM (SFP-SOM) which processed each input attribute independently on SP-SOM. In this paper, we examine the robustness of SP-SOM and SFP-SOM to the variance of input vectors with the general classification problem using iris and abalone data set and biometric authentication problem using key typing features.

## 1 Introduction

In many problems, we need to handle multi-modal vectors which are composed of multi-kinds of vectors or attributes. For example, the keystroke timing vector and key typing intensity vector are composed as the multi-modal vector in the authentication problem using key typing features. In face image classification problem, the image vector, age, gender and other features can be composed as the feature vector. In the animal classification problem which is common for SOM researches, the binary attributes which represent the size, weight, running speed and other features are integrated in a vector. In these problems, each vector or attribute is described in a different unit and scale. Additionally, the variance of each vector or attribute may be different and it affects the accuracy of classification.

For the learning of multi-modal vectors using conventional Self Organizing Map (SOM)s, the just concatenated vectors are often used. However, the resulting maps are dominated by the largely scaled vectors and are easily affected by inaccurate vectors. The effect of each vector can be adjusted by concatenating the vectors with weight values. However, the resulting maps heavily depend on the setting of weight values and it is difficult to determine optimal weight value for each vector.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 836–844, 2009.

For this problem, we proposed Pareto learning Self Organizing Map (P-SOM)s. P-SOM organizes the input data composed of the multiple independent vectors based on the Pareto optimal concept[1]. Additionally, we proposed Supervised Pareto learning SOM (SP-SOM)s to improve the accuracy of classification by adding the supervised learning of category vector as feature vectors[2]. We applied P-SOM and SP-SOM to the authentication problem using multi-modal behavior vectors such as key typing features and pen drawing features on touch screen[2]. We proposed Full Pareto learning SOM (FP-SOM) and Supervised Full Pareto learning SOM (SFP-SOM) which handle all attributes in input vector independently[3]. FP-SOM and SFP-SOM will be effective for the problems which handle the input vectors composed of the attributes of unknown features. Additionally, we reported the performance in classification problems of iris data and authentication problem using behavior characteristics, and examined the effectiveness of incremental learning of test data for both of the problems[3].

In this paper, the robustness of the Pareto learning SOMs including SP-SOM and FSP-SOM to the variances of the input vectors is reported. As the variances of input vectors, the input vectors with missing values and input vector with artificial noises are examined. P-SOMs are the multi-winner SOMs which handle the members of Pareto Set as winners. Thus, if the Pareto winner set is not so much changed by the variances of input vectors, the learning process and recalling process will not be affected so much. Especially for FP-SOMs, the input vector with missing values can be processed in the same style of standard input vectors and they will be more robust than conventional SOM and P-SOMs. Additionally, the algorithm of FP-SOMs is improved with adjusting the size of Pareto winner set adaptively. We made experiments using iris data and abalone data which are registered in UML machine learning repository and made experiments of multi-modal biometric authentication.

## 2 Pareto Learning Self Organizing Maps

### 2.1 Pareto Learning SOM

Conventional SOM can be used for integrating the multi-modal vectors. The multi-modal vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ are simply composed in a vector as follows.

$$\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) \tag{1}$$

In this case, all vectors are learned in the same weight. Considering the feature of each vector, multi-modal vectors should be composed in a vector $\mathbf{x}$ using the weight value for each vector as follows.

$$\boldsymbol{x} = (w_1\boldsymbol{x}_1, w_2\boldsymbol{x}_2, \ldots, w_n\boldsymbol{x}_n) \tag{2}$$

where $w_i$ is the weight value for vector $x_i$. Because the map is organized using the error function based on the weighted vectors, the resulting map is heavily depending on the weight values $w_i$. These weight values should be adjusted based on the scales and variances of the vectors. The normalized vectors are integrated as input vectors for adjusting the scales. In [4], each feature vector is rescaled by the standard deviation of learning data and it may help for adjusting scales and variances. However, the prior knowledge of input vectors is required for these rescaling methods.

Pareto learning SOM (P-SOM) is the SOM which is based on the concept of Pareto optimality for integrating multi-modal vectors. The Pareto set $P$ which is composed of the vectors $\boldsymbol{x}$ that are not inferior to others. For Pareto learning SOM, the input vector is given as the composition of distinct vectors as follows.

$$\boldsymbol{x} = (\{\boldsymbol{x}_1\}, \{\boldsymbol{x}_2\}, \ldots, \{\boldsymbol{x}_n\}) \tag{3}$$

The objective functions of multi-objective optimization problems are given as follows.

$$f_n(\boldsymbol{x}, U^{ij}) = \left|\boldsymbol{x}_n - \boldsymbol{m}_n^{ij}\right| \tag{4}$$

where $\boldsymbol{m}^{ij} = \{\boldsymbol{m}_1^{ij}, \boldsymbol{m}_2^{ij}, \ldots, \boldsymbol{m}_m^{ij}\}$ is the multi-modal vector which is associated to the unit $U^{ij}$ on the map. At each presentation of input vectors, the units which are in the Pareto winner set that is defined by the following equation are searched for as the winner units.

$$P(\boldsymbol{x}) = \{U^{ij} | U^{ij} \text{is Pareto optimal for the set of objective functions } f_n(\boldsymbol{x}, U^{ij})\} \tag{5}$$

Generally, the Pareto set is composed of multiple members, so P-SOM is multi-winner SOM. All winner units in $P(\boldsymbol{x})$ and their neighbors are updated simultaneously at cooperative phase. The difference between SOM and P-SOM algorithm is shown in Fig.1. The overlapped regions of neighbors are updated more strongly, and it will help the integration of multi-modal vectors on the map. P-SOM can integrate any form of vectors which use different distance metrics. Thus, the category vector can be simply integrated. P-SOM with the supervised category vectors is called as Supervised Pareto Learning SOM (SP-SOM). After learning, category of a test vector is determined by the sum of the category vectors in Pareto set of units.

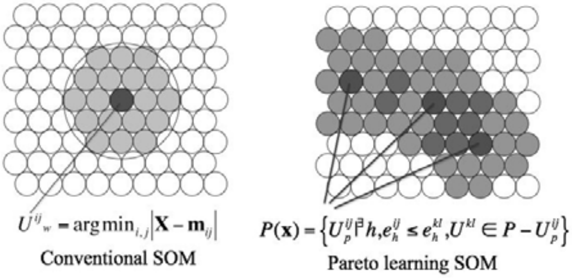

$U^{ij}{}_w = \arg\min_{i,j} \left|\mathbf{X} - \mathbf{m}_{ij}\right|$
Conventional SOM
$P(\mathbf{x}) = \left\{U_p^{ij} \nexists h, e_h^{ij} \leq e_h^{kl}, U^{kl} \in P - U_p^{ij}\right\}$
Pareto learning SOM

**Fig. 1.** Difference of the algorithm between SOM and P-SOM

## 2.2 Full Pareto Learnig SOM (FP-SOM)

Full Pareto SOM (FP-SOM) is the P-SOM which processes each attribute in the input vector independently. For input vector $(x_1, x_2, \dots, x_m)$, the objective function is given as $f_n(\boldsymbol{x}, U^{ij}) = \left| x_n - m_n^{ij} \right|$ which is the quantization error of each attribute. FP-SOM is relevant to the problem for that proper combination of input attributes is not known. Furthermore, the attributes which are described in different scales and units are processed evenly using FP-SOM. Thus, scale of each attribute does not affect to organize the map. However, the size of Pareto set becomes too large or all units on the map using FP-SOM if the number of attributes is large.

For this problem, we changed the concept of Pareto optimality to M/N Pareto optimality. A vector can be in Pareto optimal set if at least one objective function is superior to others and this condition is considered to be too weak. The M/N Pareto optimal set P is composed of the vectors which have at least M superior objective functions to those of others in P where N is the number of objective functions. However, we have another problem, which value of M is optimal?

For this problem, we propose adaptive learning method of FP-SOM. During the learning process, M of M/N Pareto optimal set is adaptively adjusted for each input vector x based on the preferred size of Pareto size for given iteration steps. The size of Pareto optimal set should be relatively large at the beginning of learning and should become smaller at the termination of learning and the value M is adaptively adjusted for the given size of Pareto optimal set in each iteration step. FP-SOM can be supervised with adding category vector and we call that as Supervised Full Pareto learning SOM (SFP-SOM). The algorithm of FP-SOM with adaptive control of Pareto size is as follows.

**FP-SOM Algorithm**

1. Initialization of the map and parameters
   Initialize the vector $\boldsymbol{m}^{ij}$ which are assigned to unit $U^{ij}$ on the map using the 1st and 2nd principal components as base vectors of 2-dimensional map. Initialize the size of neighbors Sn, learning rate *eta* and size of Pareto set Sp.
2. Batch learning phase
   (1) Clear all learning buffer of units $U^{ij}$.
   (2)For each input vector $\boldsymbol{x}^i = (\{\boldsymbol{x}_1^i\}, \{\boldsymbol{x}_2^i\}, \dots, \{\boldsymbol{x}_n^i\})$, execute the following steps.
   (2-1) Set $M = 1$
   (2-2) Search for the M/N pareto optimal set of the units $P = \{U_p^{ab}\}$. $U_p^{ab}$ is an element of pareto optimal set P, if for all units $U_{kl} \in P - U_p^{ab}$, existing h such that $e_h^{ab} \leq e_h^{kl}$ where

$$e_h^{kl} = \left| \boldsymbol{x}_h^i - \boldsymbol{m}_h^{kl} \right| \tag{6}$$

   (2-3) if $|P| \leq Sp$ goto (3), else M=M+1 and goto (2-1)
   (3) Add $x^i$ to the learning buffer of all units $U_p^{ab} \in P$.

3. Batch update phase
   For each unit $U^{ij}$ update the associated vector $\mathbf{m}^{ij}$ using the weighted average of the vectors recorded in the buffer of $U^{ij}$ and its neighboring units as follows.
   (1)For all vectors $x$ recorded in the buffer of $U^{ij}$ and its neighboring units in distance $d \leq Sn$, calculate weighted sum $\mathbf{S}$ of the updates and the sum of weight values W.

$$\mathbf{S} = \mathbf{S} + \eta fn(d)(\mathbf{x} - \mathbf{m}^{\mathbf{i'j'}}) \tag{7}$$

$$W = W + fn(d) \tag{8}$$

   where $U^{i'j'}$s are neighbors of $U^{ij}$ including $U^{ij}$ itself, $\eta$ is learning rate, $fn(d)$ is the neighborhood function which becomes 1 for d=0 and decrease with increment of d.
   (2) Set the vector $\mathbf{m}^{\mathbf{ij}} = \mathbf{m}^{\mathbf{ij}} + \mathbf{S}/W$.

Repeat 2. and 3. with decreasing Sn, $\eta$ and Sp for pre-defined iterations.

## 3 Experimental Results of Robustness to Variances of Input Vectors

### 3.1 Iris Classification Problem

In this section, the experimental results of robustness to variances of input vectors using Supervised Pareto learning SOMs are mentioned. As the classification problem, the iris data set obtained from UCI machine learning database are used. The iris data set contains 3 classes of 50 instances of each, where each class is refers to a type of iris plant. each datum contains 4 attributes, which are sepal length(sl), sepal width(sw), petal length(pl) and petal width(pw) in cm. We reported that the accuracy of the classification depended on the combinations of the attributes to multi-modal vectors[3]. In this paper, we show the results of experiments in 2 cases.

- Case 1: $\boldsymbol{x} = (sl, sw, pl, pw)$ All attributes are composed in a vector.
- Case 2: $\boldsymbol{x} = (\{sl, pl\}, \{sw, pw\}$ 2 vectors which represents the features of length and width respectively are integrated.

Case 1 is almost identical to conventional SOM using the category vector for supervised learning, so we use this classification result as that of conventional SOM. For each case, 50 iterations of experiments are made with changing the number and combination of learning data and test data. ?

### 3.2 Experimental Results for Input Vector with Missing Values

In this section, we made experiments using input vector with missing values. As the missing values, we used input vector with omitting one of the attributes

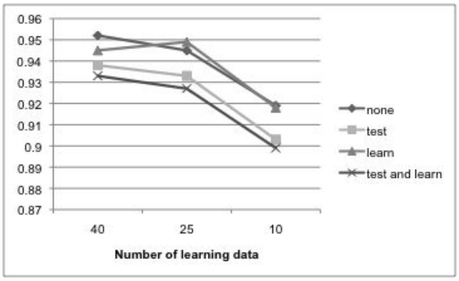


**Fig. 2.** The results input vectors with missing values in case 1 of iris

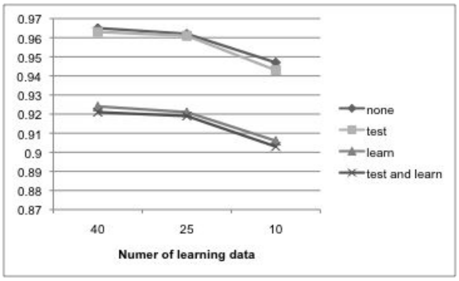


**Fig. 3.** The results for input vector with missing values in case 2 of iris

from 4 attributes. SFP-SOM can handle missing values without changing algorithm. However, the modification of the algorithm is needed for these cases. We applied the method reported in [5] to handle missing values. Fig.2 and Fig.3 show the result for case 1 and case 2 respectively. In these figures, none, test and learn denote that none of the vector is omitted (all attributes are used), missing values in test vector and missing values in learning vector, respectively and X-axis denotes the number of learning data. In case 1, missing values in learning vectors do not affect the accuracy, however, missing values in test vectors degrade the performance. Inversely, in case 2, missing values in test vectors do not affect the accuracy, however, missing values in learning vectors degrade the performance. It is considered that SP-SOMs are robust to the missing values in test vectors because the changes of the Pareto set are in the acceptable range, however, variation of the Pareto set with missing values in learning vectors cannot be interpolated. Inversely, conventional SOM can interpolate missing values in learning vectors, however, the differences of quantization errors using test vectors with missing values are not acceptable.

### 3.3 Experimental Results for the Input Vector with Noise

In this section, we made experiments using the input vector with artificial noises. As the input vector with noise, we add maximum ±50% amplitude of random noise to one of 4 attributes. Fig.4 and Fig.5 show results for case 1 and case 2 respectively. For all cases, the results are more affected for the case with the input vector with noise in the test than for the case with the input vector with noise in learning. Among these cases, case 2 is less degraded. From these results, SP-SOMs are considered to be more robust to the noises in input vectors and they are rather more robust to the noises in learning vectors.

On the other hand, SP-SOM shows less decreasing rate in response to the decreasing of the number of learning data for all cases of missing data and noises. SP-SOM is considered to be also more robust to the variation of the number of learning data.

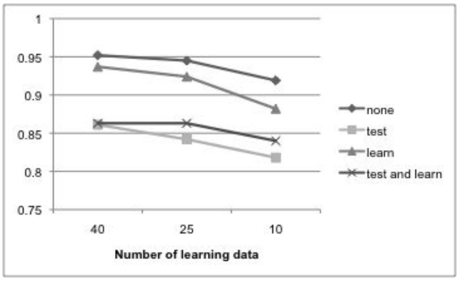


**Fig. 4.** The results for input vectors with noise in case 1 of iris

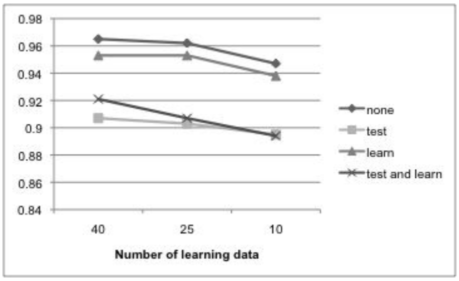


**Fig. 5.** The results for input vectors with noise in case 2 of iris

### 3.4 Experimental Results for Abalone Data

The number of attributes of iris data is considered to be too small for testing Full Pareto learning SOMs. In this section, the experimental results for abalone data which are also registered in UCI machine learning database are shown. The abalone data contain 4177 data, and each datum contains 9 attributes. One of the attributes denotes the number of rings of the abalone, and it is classified in 3 classes, less than 9, 9 or 10, and 11 on in benchmark. In the document attached to the data, 3133 data are used for learning and 1044 remainders are used for test. As the combination of multi-modal vectors, we made experiments in 2 cases. In case 1, all attributes are composed in a vector (conventional SOM) and in case 2, all attributes are composed as independent vectors (SFP-SOM). As the variances of input vectors, the input vectors with noises are used. Fig.6 and Fig.7 show the results. In case 1, the accuracy increases with decreasing the number of learning data. In case 2, the accuracy is the best with the number of learning data 2577 and overall accuracy is superior to those of case 1. As for the noises, the noises in the test vectors affect more than that in learning vectors for both cases. We made other experiments with changing the combinations of multi-modal vectors using SP-SOM and SFP-SOM marked the best accuracy of classification.

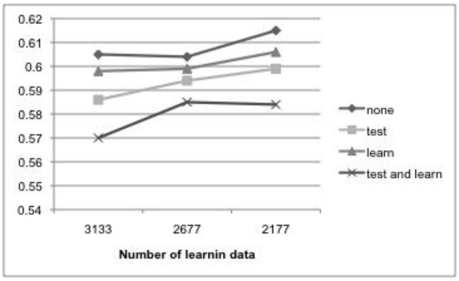


**Fig. 6.** The results for input vectors with noise in case 1 of abalone data

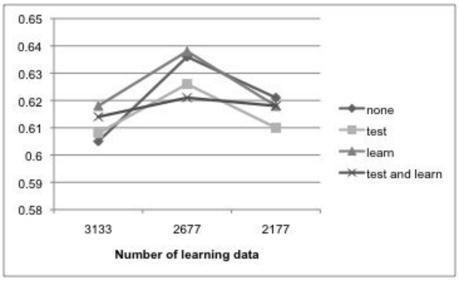


**Fig. 7.** The results for input vectors with noise in case 2 of abalone data

### 3.5 Experimental Results for Multi-modal Biometric Authentication

We reported the application of SP-SOM to authentication problem using the keystroke timing and key typing sound as the multi-modal behavior biometrics[3]. The keystroke timing data are the vectors composed of the intervals of the pushing and releasing the keys. The key typing sound data are the vectors composed of the maximum amplitudes of the sounds in each key typing. From each of 10 examinees, 10 data typing the same phrase "kirakira" are sampled. In this paper, the authentication experiments are made with adding the artificial noises to the test vectors used for authentication. The map is learned using one registration vector from 10 samples and authentication experiment is made using 9 remainder samples. As the combination of multi-modal vectors, 2 cases are examined. In case 1, the multi-modal vector is composed from the vector of keystroke timings and the vector of key typing sounds and processed using SP-SOM. In case 2, all attributes of keystroke timings and key typing sounds are independently processed as vectors using SFP-SOM. The noises are added to the input vector in authentication test with probability P=0.25 or P=0.5 for each attribute and the amplitude of the noise is ±50% random. Fig.8,9,10 shows the results. The horizontal axis of these graphs denotes TAR(True Acceptance Rate) which is the rate for successful authentication of true user. Comparing case 1 and case 2, TARs are better and variances among the users are smaller

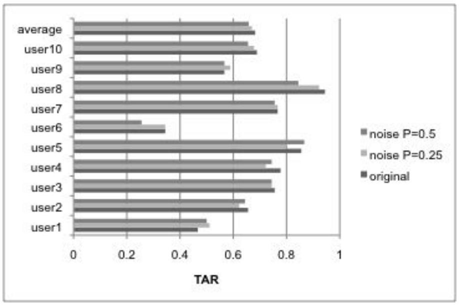


**Fig. 8.** The results of authentications for input vectors with noise in case 1

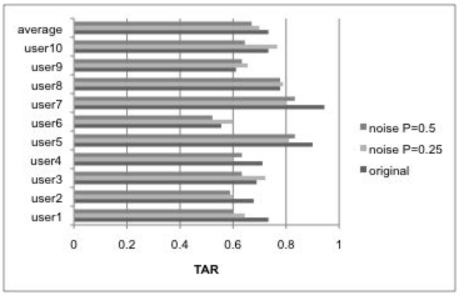


**Fig. 9.** The results of authentications for input vectors with noise in case 2

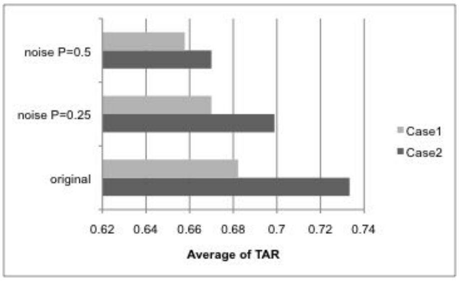


**Fig. 10.** Comparison of case 1 and case 2

in case 2 using SFP-SOM. As for noises, the authentication results are slightly affected considering that probability and amplitude of the noises are high. In these cases, SFP-SOM marks better performance than that of SP-SOM, however in other cases changing the number of registration data, SP-SOM marks the better result because the number of attributes(23 in this problem) is too large for processing much number of learning data using SFP-SOM. The algorithm of SFP-SOM should be improved to solve this problem.

## 4 Conclusion

In this paper, we reported the robustness of the Pareto learning SOMs to the variances of input vectors. Pareto learning SOMs are robust to the variance of test input vectors for missing values in test vectors and noise in learning vector. Full Pareto learning SOM shows superior performance for classification problem without prior knowledge of the combination of multi-modal vectors.

As the future works, the algorithm of P-SOMs, especially FP-SOMs, should be improved. In many cases, the integration of multi-modal vector using conventional SOM whose weight value for each element vector is tuned carefully with iteration of experiments shows better performance than that of SP-SOM. We should improve SP-SOM and FSP-SOM algorithm to be superior or same performance without tuning of parameters. Furthermore, anything which has its own metrics can be the element of multi-modal input vector for P-SOM and SP-SOM. Structured data and vectors can be integrated as input data. For example, using the sequence data and its features of DNA sequences as input vector, the units can be represented using Hidden Markov Model for sequence data and independent vectors for each feature. We should examine the performance of P-SOM and SP-SOM in a variety of applications.

## References

1. Dozono, H., Nakakuni, M., et al.: Application of Self Organizing Maps to User Authentication Using Combination of Key Stroke Timings and Pen Calligraphy. In: Proceedings of the 5th WSEAS Int. Conference on Computational Inteligence, pp. 105–110 (2006)
2. Dozono, H., Nakakuni, M., et al.: An Integration Method of Multi-Modal Biometrics Using Supervised Pareto Learning Self Organizing Maps. In: Proceedings of the Internal Joint Conference of Neural Network 2008 (2008)
3. Dozono, H., Nakakuni, M., et al.: Application of Supervised Pareto Learning Self Organizing Maps and Its Incremental Learning. In: Príncipe, J.C., Miikkulainen, R. (eds.) WSOM 2009. LNCS, vol. 5629, pp. 54–62. Springer, Heidelberg (2009)
4. Takatsuka, M., Javis, R.A.: Encoding 3D structural information using multiple self organizing feature maps. Image and Vision Computing 19(3), 99–118 (2001)
5. Sorjamaa, A., et al.: Sparse Linear Combination of SOMs for Data Imputation: Application to Financial Database. In: Príncipe, J.C., Miikkulainen, R. (eds.) WSOM 2009. LNCS, vol. 5629, pp. 290–297. Springer, Heidelberg (2009)

# Interactive Hierarchical SOM for Image Retrieval Visualization

Yi Liu and Masahiro Takatsuka

ViSLAB, The School of IT, The University of Sydney, NSW 2006 Australia

**Abstract.** This paper presents an interactive hierarchical visualization system for an image retrieval application. This visualization system needs to present the similarities of images. Furthermore, it is required to provide an easy way to explore and navigate images' feature space at different levels of detail. Our system utilizes a Multi-layer Geodesic Self-Organizing Map (GeodesicSOM) and Learning Vector Quantization (LVQ) to increase the accuracy in data representation at different levels of detail. The Multi-layer GeodesicSOM provides fast access/navigation to a large amount of image data while the LVQ rectifies the inconsistency in topological data representation between different layers.

**Keywords:** Content-based Image Retrieval, Information Visualization, Hierarchical Self-organizing Map, Learning Vector Quantization.

## 1 Introduction

Information retrieval is the process of gathering information such as files, web pages and images using keywords and attributes. It now plays an integral role in the daily lives of the world's workforce. Having used traditional text based retrieval for a long time, people have realized the limitations of search-by-keywords and demanded more advanced approaches to search for images and videos. Content-based Image Retrieval(CBIR) methods have been developed to retrieve images by their visual attributes such as colors and shapes.

Providing of a good visualization of search results also helps people find desired information more efficiently. However, visualization of the results of CBIR is still a new area of research. The challenge for the visualization of Image Retrieval is to preserve images' similarity relationships as much as possible, while providing an easy-to-use user interface to rapidly explore and navigate the high-dimensional feature space.

Self-Organizing Map (SOM) is one of very popular algorithms for generating pictorial representations of complex (typically highly nonlinear and high-dimensional) data spaces. It has been used to fulfill the visualization task in information retrieval applications[1,2].

Datasets typically targeted in information retrieval are usually very large. Some of them also have inherently hierarchical data structures (such as newspaper articles, DNA databases and images/videos with context-based categories).

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 845–854, 2009.

In order to cope with the large and hierarchical nature of data, many hierarchical SOMs have been developed [3,4,5,6,7].

These hierarchical SOMs, however, present a problem when they are used for navigating large datasets between different levels of detail. They fail to preserve the hierarchical topological relations (parent-child relations) between layers as explained in Section 3. This would cause the loss of the search/tracking path when a user explores certain images through different layers. In order to address this issue, our system provides geodesic multi-resolution hierarchical structured neural maps. It carries out two tasks: unsupervised topological mapping based on SOM, and supervised learning vector quantization to rectify the inconsistencies of hierarchical topological relationships.

The remainder of this paper is organized as follows. Section 2 presents an overview of closely related techniques, followed by Section 3, which states the problem to be addressed. Section 4 introduces the proposed hierarchical SOM with a novel training procedure. Experimental results are given in section 5, followed by the conclusion in Section 6.

## 2 Related Works

### 2.1 Hierarchical Feature Maps

The key idea of hierarchical feature maps as proposed in [4] is to use multiple layers of SOMs, which are different in sizes for representing data with different levels of details. The highest level is a single SOM with rather small number of units. The training process starts at the root layer. When the root layer is stabilized, the training proceeds to the next layer. This structure allows significant computational reduction when a user drills down the hierarchy to find the information.

According to the hierarchical structure employed in this, all the subcategorizations in submaps fully depend on the categorization in the preceding layer. Each map in the lower hierarchy forms subcategorizations that only map the differences within the category defined in the previous layer.

### 2.2 Tree-Structured Self-Organizing Map (TS-SOM)

Tree-Structured SOM was introduced by Koikkalainen and Oja in 1990[5]. It consists of a number of SOMs organized in a quadtree structure. The neurons are the nodes of the tree, and each layer forms a SOM. Unlike the Hierarchical Feature Maps, TS-SOM has one SOM for each layer. The training procedure starts from the root, and carries out the lateral search at each layer to find the local Best Matching Unit (BMU) and proceeds to the lower layer. The tree-structured search algorithm reduces time complexity of the search from $O(N)$ to $O(\log N)$. Additionally, it introduces tolerance to upper layers' clustering capability. Because TS-SOM's quadtree structure pre-determines the mappings between the parent node and its child nodes, it is prone to have data representation inconsistency between different layers.

### 2.3 Multilayer Self-Organizing Feature Map (MLSOFM)

MLSOM was introduced by Kon et al.[6] to solve a Range Image Segmentation problem in 1995. Similar to TS-SOM, it consists of multiple competitive layers, and each competitive layer is comprised of a SOM. The number of neurons decreases at successive levels from bottom to top, resulting in a pyramidal structure. Unlike other hierarchical SOM algorithms, the training procedure starts at the bottom layer. The trained weights of the bottom layer neurons are fed to the upper layer as new input vectors. This process repeats until the top layer is reached.

Since the MLSOFM trains itself in a bottom-up order, and the upper level clustering is based on the categorization of input data in the lower layer, the parent-child nodes are intrinsically connected. Thus, this parent-child connection adapts better to the topology of the input, thereby gaining a smaller chance of losing track of the hierarchical path during the exploratory visualization.

## 3 The Problem of Inconsistent Data Representation

The primary purpose of hierarchical SOMs was to reduce computational complexity and cost[3,4,5]. They have been successfully used in various applications (e.g.,[2,8]). The basic differences among them are their hierarchical structures and training mechanisms. However, they all seem to present the following problem of inconsistency in data representation between layers.

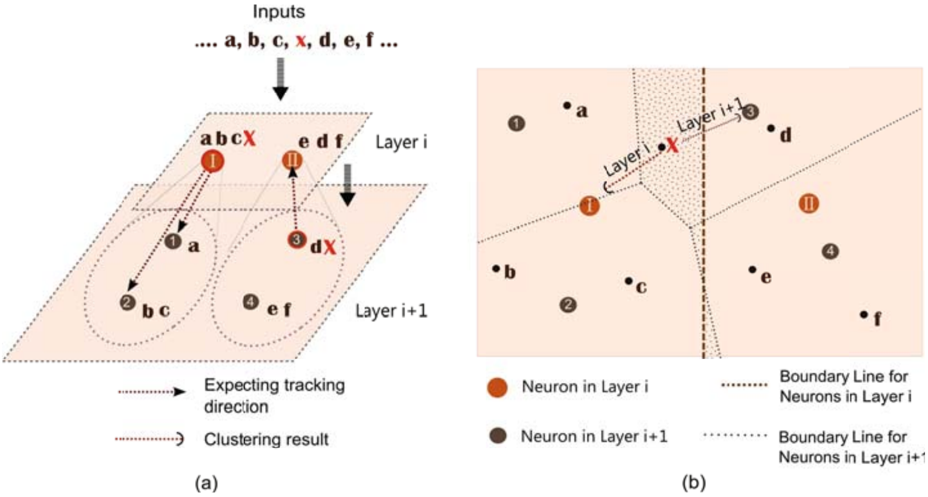


**Fig. 1.** Data representation of inconsistency in a hierarchical SOM. (a) An input data $X$ will be lost when tracking from node I in layer $i$ down to its child nodes in layer $i+1$, and so will it happen with tracking from neuron 3 up to its parent neuron II. (b) illustrates how would this issue happen in 2D space.

Figure 1(a) demonstrates how this problem surfaces in hierarchical SOMs, and Figure 1(b) illustrates the inconsistency of class boundaries, that causes this problem. Figure 1(a) shows two successive layers in a hierarchy. In this figure, the hierarchical connections between nodes are: the nodes I and II are the parents of child nodes $1, 2$ and $3, 4$ respectively. $X$ represents an input datum that is represented by the neuron I in layer $i$, but is represented by neuron 3 in layer $i + 1$, which is not a child of the neuron I. Therefore, when tracking $X$ from layer $i$ down to layer $i+1$ in I's child nodes, you would lose it. In a similar manner, when zooming out from layer $i+1$ up to layer $i$, data $X$ will be missed at the higher layer.

Figure 1(b) illustrates how this inconsistency occurs around the class boundaries. Points $a, b, .., f$ including $X$ represent the positions of the input vectors. Neurons in layer $i$ and layer $i + 1$ are represented respectively by circles I, II and circles $1, 2, 3, 4$. As shown in the figure, data $X$ is located closer to neuron I than II in the upper layer. However, it is closer to the neuron 3 in the lower layer than any of neuron I's child neurons 1 and 2. This problem will exist as long as Voronoi boundaries do not match between different layers.

## 4 Interactive Hierarchical Geodesic SOM

### 4.1 Overview of the Image Retrieval System

Our Image Retrieval Visualization system uses the proposed Hierarchical Geodesic-SOM to mediate the search process as shown in Figure 2. The SOM is trained using the two stage training method described in this section. The trained SOM generates the visualization of images which are relevant to user's interests. A user can interactively specify an image and retrieve its relevant images through the Hierarchical GeodesicSOM by navigating through different levels of the hierarchy.

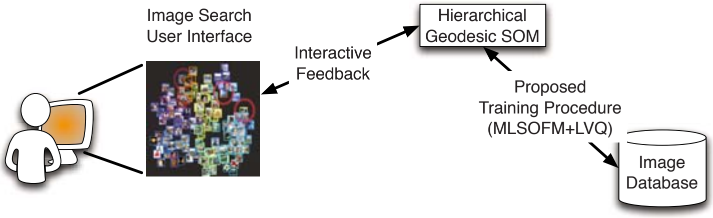


**Fig. 2.** Overview of the image retrieval system

### 4.2 Hierarchical GeodesicSOM

The proposed hierarchical GeodesicSOM consists of multiple Geodesic SOMs[9] and employs a training process similar to that of MLSOFM. The training procedure starts at the bottom layer. The input data are first fed into the bottom layer. Let $W_1 = w_1, w_2, ..., w_N$ be the set of weight vectors at layer 1, where $N$ is the number of neurons in this layer. On completion of training, $W_1$ represents the probability density function of the input data. Unlike MLSOFM[1], all weights in $W_1$ will then be converted to the input vectors for training in layer 2, and the neurons in layer 2 are initialized with the weights of their geometrical child nodes based on the hierarchical geodesic grid structure. This procedure repeats itself until the top layer (layer m) is reached. The external input data has only been used to train the bottom layer, the layers above are trained based on neurons in their immediate lower layer.

### 4.3 Hierarchical Weight Adjustment for Maintaining the Parent-Child Connection

As described in Section 3, the use of hierarchical SOMs could cause inconsistent data representation between layers. In order to address problem, this work proposes the second training stage to rectify this inconsistency. Our proposed weight adjusting algorithm is based on the Learning Vector Quantization (LVQ) algorithm. Since the LVQ is a supervised learning algorithm, it requires teaching signals. Because the purpose of this process is to rectify data representation inconsistency between parent and child neurons, we adjust weights of neurons based on their parent-child relationships.

The process starts from the second top layer (layer $m-1$), and ends at the bottom layer (layer 1). This time, we feed the real data to each layer for learning. For each input $x$, $w_i$ represents the weight vector of the BMU in layer $(i)$ and $w_{i-1}$ represents the weight vector of the BMU in the layer above (layer $i-1$). At each time step $t$, the map is updated in the following manner:

$$w_i(t+1) = w_i(t) + \alpha(t) \cdot \lambda \cdot [x - w_i(t)],$$

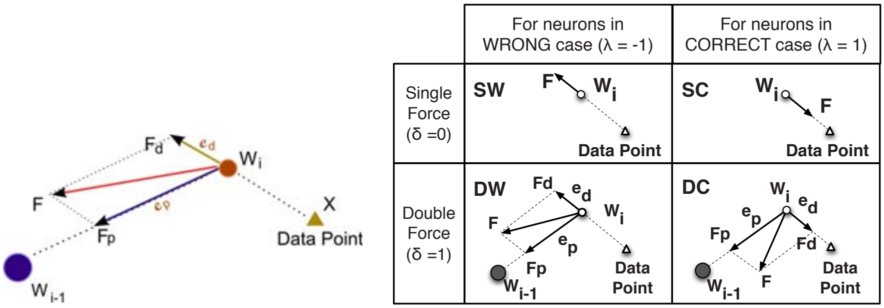
$$\lambda = \begin{cases} +1 & : \text{if } w_i(t) \in C(w_{i-1}(t)), \\ -1 & : \text{if } w_i(t) \notin C(w_{i-1}(t)), \end{cases} \tag{1}$$

where $C(w_{i-1}(t))$ denotes the set of children of $w_{i-1}(t)$; $\alpha(t)$ $(0 < \alpha < 1)$ is the learning rate at time $t$, and it is made to decrease monotonically with time.

This is the basic version of our LVQ weight adjusting algorithm, which considers only data influence (Single Force cases in Figure 3(b)). For simplicity, we refer to a case, where an input $x$ is represented by the two BMU weights ($w_i(t)$ at the layer $i$ and $w_{i-1}(t)$ at the layer $i-1$) in the parent-child relationship, as a “CORRECT” case. We will call a case, where an input $x$ is represented by the two non parent-child BMU weights, as a “WRONG” case. In single force cases, neurons that are in the “CORRECT” case receive further encouragement to update weight vectors towards assigned data, while neurons are in the “WRONG” case receive the weights update to move them away from its assigned data.

As also could be observed from the Voronoi Diagram of neurons, the wider the child nodes scatterd, the higher probability that “WRONG” cases could

[1] The MLSOFM only converts the neurons that have data assigned into inputs.



(a) Two forces used for LVQ algorithm. $F = F_d + F_p$

(b) Combinations of updating forces

**Fig. 3.** Combinations of the number of forces and their directions used in the LVQ process. SW: a single force on a wrong case, SC: a single force on a correct case, DW: double forces on a wrong case, DC: double forces on a correct case.

occur. Therefore, we also proposed an advanced version of this function, which considers not only data influence but also parent influence (Double Force Cases in Figure 3(b)).

$$w_i(t+1) = w_i(t) + \alpha(t) \cdot \lambda \cdot e_d[x - w_i(t)] + \alpha(t) \cdot \delta \cdot e_p[w_{i-1}(t) - w_i(t)],$$

$$\delta = \begin{cases} 1 & : \text{if the parent influence is considered} \\ 0 & : \text{if the parent influence is not considered} \end{cases} \tag{2}$$

$\delta$ is a flag factor that determines if the parent neuron influence will be added or not, which will be covered later in this section; $e_d$ and $e_p$ are the weighting factors on the two type of forces, which will be further discussed in section 5.

The weight update is governed by two types of forces, data-influence force ($F_d$) and parent-influence force ($F_p$), as shown in Figure 3(a). Data-influence force, $F_d$ represents the updating force from the BMU ($w_i(t)$) to the input $x(t)$. Parent-influence force $F_p$ indicates the updating force from the BMU ($w_i(t)$) to the parent BMU[2] ($w_{i-1}(t)$). As a result, neurons that are in the "CORRECT" case receive further encouragement to update weight vectors. On the other hand, if neurons are in the "WRONG" case (misrepresenting data between the parent and child), they receive the weights update to move them away from its assigned data, and move towards its parent BMU.

Depends on whether to add the force from the parent neuron influence, we have four combinations of ($\lambda$, $\delta$) (as shown in Figure 3(b))for experiments, and the results are presented in Section 5.

## 5 Experimental Results and Discussion

We evaluated the performance of the proposed method with respect to the hierarchical path inconsistency in the interactive image retrieval application with

[2] Parent BMU refers to the winner of $x(t)$ in the previous layer (layer $i - 1$).

two different image data sets. The first data set used in this experiment contains 400 images that were found using the Yahoo API. It consists of 10 categories, (for example, flowers, bays, etc.) and each has 40 images.

The second data set is from the WANG data set that can be obtained from the web site [10]. It is a subset of the Corel image collection and contains 1000 images. This data set consists of 10 categories, such as images of Africa, buildings, elephants, etc. A 25 dimensional CompactCCM feature, which is one of the image color features has been extracted from all image data sets for testing.

A three layer hierarchical GeodesicSOM was used for the experiment. The first layer uses a 1st-frequency geodesic dome. Consequently the second layer uses a 2nd-frequency geodesic dome and the third layer uses a 4th-frequency geodesic dome. The number of neurons $numU$ in $f^{th}$-frequency geodesic dome at layer $i$ can be calculated as: $numU = 2^{f-1} * 10 + 2$ Hence, the numbers of neurons in the three layers of the hierarchical GeodesicSOM are 12, 42 and 162, from top to bottom. During the Hierarchical SOM training, each map has gone through 800 iterations, and the same number during the LVQ adjustment process. The initial learning rate for Hierarchical SOM training was set to 0.8, and 0.5 for the LVQ adjustment.

We also performed the TS-SOM algorithm and MLSOFM algorithm with the same training parameters on the same data set for comparison. After the training process had been completed, we calculated the error rate, indicating the number of images whose winners in adjacent layers are not in a correct parent-child relation. The values given in the results figure are the percentage of the amount of these images with inconsistent parent-child relationships. Moreover, there are four distinct cases in the combination of updating forces during the LVQ process as shown in Figure 3(b). In the results shown below, the notation of SW, SC, DW and DC were used to identify which updating force combinations were used.

Results based on the Yahoo data set are shown in Figure 4, and results based on the WANG data set are shown in Figure 5. Comparison error rates results between different hierarchical SOM algorithms and different forces combinations

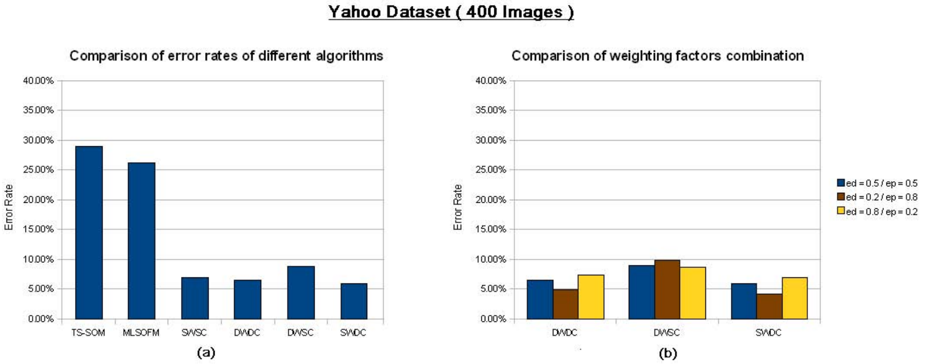


**Fig. 4.** Hierarchical consistency error rates of Yahoo Dataset

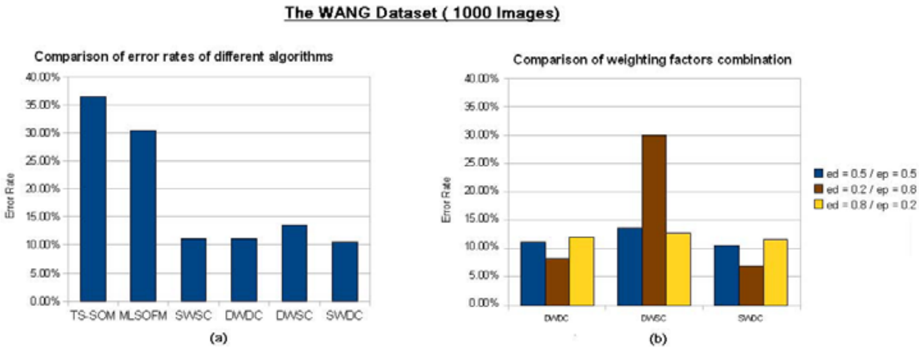
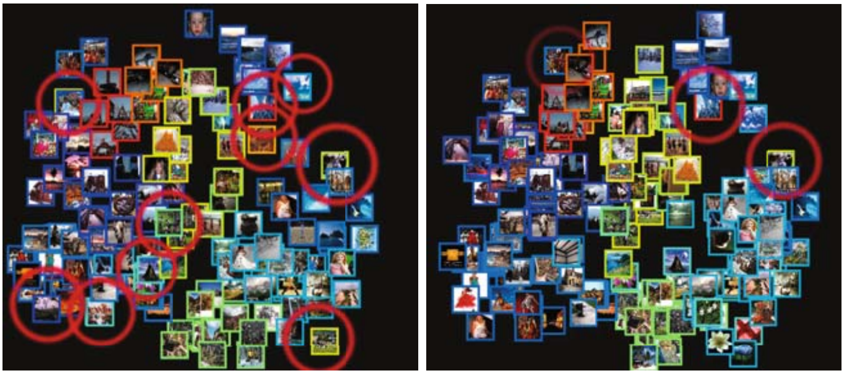


**Fig. 5.** Hierarchical consistency error rates of WANG Dataset

of proposed methods are given in Figure 4(a) and Figure 5(a). As shown in Figure 4(a), the error rates for TS-SOM and MLSOFM were 29.00% and 26.25% respectively and scored the worst and 2nd worst results. On the other hand, the proposed method yield very low error rates (7.00% for SWSC, 6.50% for DWDC, 8.88% for DWSC and 5.88% for SWDC).

Although our approach produced much lower error rates, there seem to be differences in the combinations of updating forces used during the LVQ process. We have tested the different combinations of updating forces with different weighting factors on each force. We varied the weighting factors of the updating forces between 0.2, 0.5, and 0.8 in order to check the effect of forces from the parent neuron and the input datum. The results of this comparative analysis are



(a) without boundary rectification (b) with boundary rectification

**Fig. 6.** Screenshots of a user interface visualizing clustering of images produced by the hierarchical SOM. The red circles in the image indicate the inconsistent data representations between different layers in the hierarchy.

shown in Figure 4(b) and 5(b). The results shown in Figure 4(b) indicated that the use of two forces ($F_d$ and $F_p$) for the correct case with $e_d = 0.2$ and $e_p = 0.8$ effectively removed the inconsistency.

In order to demonstrate how this hierarchically inconsistent path affects the visualization interface of the image retrieval system, screenshots of the visualization are shown in Figure 6. The visualization system places images according to their similarities computed by the hierarchical SOM. Different colored borders were used to indicate which images share the same parent neuron in the higher layer. Images which do not share the same parent neuron with nearby images are circled in red. Figure 6(a) is the result without hierarchical inconsistency being rectified by the LVQ process. It is clear that the number of miss-paring is significantly reduced by rectifying the hierarchical inconsistency using the LVQ.

## 6 Conclusion

In this paper, we proposed a new hierarchical SOM-based visualization system for image retrieval application. We first described the issue of inconsistent data representation between different layers in typical hierarchical SOM solutions. Based on the hierarchical structure and the bottom-up training process in the MLSOFM, a Multi-layer GeodesicSOM was constructed. In order to rectify the inconsistent data representation between layers, the LVQ algorithm was employed to adjust the position of neurons. Experimental results showed that the introduction of the new system led to a significant reduction of the error in data representation between layers.

## References

1. Honkela, T., Kaski, S., Lagus, K., Kohonen, T.: Websom - self-organizing maps of document collections. Neurocomputing, 101–117 (1997)
2. Laaksonen, J.T., Koskela, J.M., Oja, E.: Picsom - a framework for content-based image database retrieval using self-organizing maps. In: 11th Scandinavian Conference on Image Analysis, pp. 151–156 (1999)
3. Luttrell, S.: Hierarchical self-organising networks. In: First IEE International Conference on Artificial Neural Networks (Conf. Publ. No. 313), October 1989, pp. 2–6 (1989)
4. Miikkulainen, R.: Script recognition with hierarchical feature maps. Connection Science 2, 83–101 (1990)
5. Koikkalainen, P., Oja, E.: Self-organizing hierarchical feature maps. In: IJCNN International Joint Conference on Neural Networks, June 1990, vol. 2, pp. 279–284 (1990)
6. Koh, J., Suk, M., Bhandarkar, S.M.: A multilayer self-organizing feature map for range image segmentation. Neural Netw. 8(1), 67–86 (1995)
7. Dittenbach, M., Merkl, D., Rauber, A.: The growing hierarchical self-organizing map. In: Proceedings of the International Joint Conference on Neural Networks, vol. VI, pp. 15–19. IEEE Computer Society Press, Los Alamitos (2000)

8. Kohonen, T., Kaski, S., Lagus, K., Honkela, T.: Very large two-level som for the browsing of newsgroups. In: Vorbrüggen, J.C., von Seelen, W., Sendhoff, B. (eds.) ICANN 1996. LNCS, vol. 1112, pp. 269–274. Springer, Heidelberg (1996)
9. Wu, Y., Takatsuka, M.: Spherical self-organizing map using efficient indexed geodesic data structure. Neural Networks 19(6-7), 900–910 (2006)
10. Wang, Z.: James, J.z.wang's research group. online resource, http://wang.ist.psu.edu/

# Evaluation Patterns of Japanese Representative Athletes in the 2008 Beijing Olympic Games: Visualization of Social Expectation and Satisfaction by Use of Self-Organizing Maps

Tetsuya Onoda

School of Information-Oriented Management, SANNO University
1573, Kami-Kasuya, Isehara, Kanagawa 259-1197, Japan
ond@ond-lab.net

**Abstract.** The purpose of this research is to establish the model of quantitatively capturing the changes in time series of a domestic public opinion toward the Japanese representative in international sports events for the top athlete management. Beijing Olympics, held in August 2008, was thought to be one of the best cases to carry out this approach. As a result, extracting six evaluation patterns by comparing the social involvement levels between opening and closing of the Olympic games has been succeeded by use of Self-Organizing Maps.

**Keywords:** Self-Organizing Maps, Athlete Management, The 2008 Beijing Olympic Games, Expectation-Confirmation/Disconfirmation Model, The Long Tail.

## 1 Introduction

The purpose of this research is to establish the model of capturing the changes in time series of a public opinion toward athletes. This paper is based on the 2008 Beijing Olympic Games, which was held from August 8th to 24th and it focuses the domestic public opinion toward the Japanese representative athletes.

The most important framework this research refers, is Expectation-Confirmation/ Disconfirmation Model [1][2]. This model is usually applied to figure out the mechanism of the consumer satisfaction process based on the involvement gap between “expectation” before purchase and “satisfaction” after that. The reason why the importance of this model is emphasized is because it can be considered to be the similar framework if we transpose “consumer” to “spectator” and “product” to “athlete”.

However, this research has the novelty about the application of the model. While the original model deals with consumers as individuals, this new model deals with spectators as holistic Japanese society. Therefore public opinion survey was carried out. In addition, there is another uniqueness on this research. It is referring the Long Tail [3] in the marketing field. If we depend on the new idea for the product management, we should survey not only famous “Head” athletes but also unknown athletes as the “Tail”. Therefore this research launched every 339 representative athletes into the survey sheet.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 855–864, 2009.

## 2 Method

Soft-Structuring Model (SSM) [4] is applied to this research. It is the model which achieved satisfactory results in some other fields. This model has five processes and Self-Organizing Maps (SOM) [5] is included in the 4th process. Concrete explanations of each processes are the following.

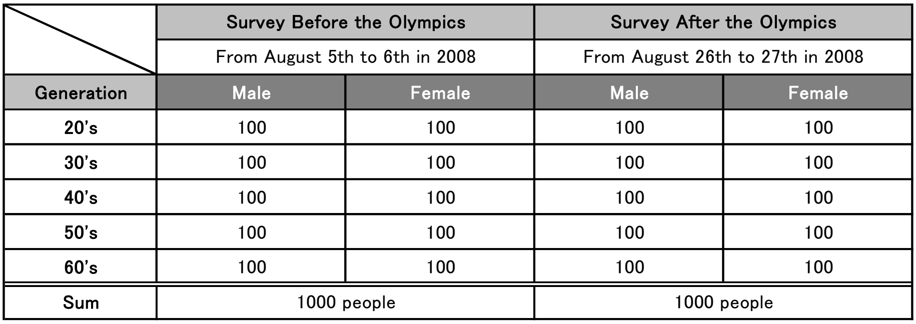
### 2.1 Internet Survey

As mentioned above, every representative athlete of Japan is mounted on surveys. The list of these athletes can be all collected by referring to the official website of the Japanese Olympic Committee (JOC) [6].

However, it had two problems. One was the time of final certification. It was 24th of July when mere two weeks remained for the opening of the Beijing Olympics. The other problem was the huge number of athletes. It was too difficult to apply conventional way of survey, such as interview or mail survey. Therefore in this case, the internet survey was thought to be the best survey method because huge amount of items could be launched in a short time and respondent could answer easily by clicking of mouse BIOS.

The internet survey performed twice for each 1000 samples of 10 sex and age demographics which are shown in Table.1.

**Table 1.** Survey Term and Composition of the Respondent



| | Survey Before the Olympics | | Survey After the Olympics | |
|---|---|---|---|---|
| | From August 5th to 6th in 2008 | | From August 26th to 27th in 2008 | |
| Generation | Male | Female | Male | Female |
| 20's | 100 | 100 | 100 | 100 |
| 30's | 100 | 100 | 100 | 100 |
| 40's | 100 | 100 | 100 | 100 |
| 50's | 100 | 100 | 100 | 100 |
| 60's | 100 | 100 | 100 | 100 |
| Sum | 1000 people | | 1000 people | |

### 2.2 Data Correction

As discussed previously, this research approach has the same framework as Expectation- Confirmation/Disconfirmation Model. At the same time, the Long Tail is emphasized as well. In such a case, we have to pay attention to the respondent because it is unrealistic for them to answer both "expectation" and "satisfaction" on all 339 athletes, even though by mouse click. It is necessary to fix the "acknowledgement" question before asking "expectation" to narrow down the number of the athletes. At the same time, we have to fix the "spectator" question before asking "satisfaction".

Therefore four input variables, (A)Acknowledgement, (B)Expectation, (C)Spectator, and (D)Satisfaction, are use to analyze. The survey to acquire four variables was conducted twice. One is the survey before the Olympics for (A) and (B), the other is the survey after that for (C) and (D). However four variables are divided into two types from the perspective of the data characteristic. (A) and (C) are 1/0 flag data, so we can treat the summary of them as percentages. On the other hand, (B) and (D) are ordinal scale data as Table.2, so we will provide the score, fitting each choice, and the figure of the summaries are the total.

Normalization of the scale is necessary to analyze two different types of summary data in one frame. Although there are several kinds of normalization method, the Cumulative Probability of sampling Normal Distribution of each attribute is adopted in this research. It is not better to exit negative values in the next process of SSM. One of the reasons of this adaptation is because this probability ranges from 0 to 1.

**Table 2.** Correspondence between Choices and Scores on (B)Expectation and (D)Satisfaction

| Score | Choices of (B)Expectation | Choices of (D)Satisfaction |
|---|---|---|
| 5 | I will definitely check the broadcasting time not to miss the game. | Exceed Expectation |
| 3 | I will watch the game curiously if the game is on television. | As Expected |
| 1 | If the athlete is doing well in the game, I will not change the program. | Acceptable |
| 0 | I have no reason to watch the game. | Disappointed |

## 2.3 Creating Layers

Fig.1 as the Long Tail is the descending order graph of the summation of normalized values of (A), (B), (C) and (D). This graph shows two problems for the next analysis. The first problem is that extreme "Tail" is included. The data of these athletes are too unstable to analyze. The second problem is the huge absolute differential exits between "Head" and "Tail". To solve these problems, layers are needed to be created.

The Cumulative Probability of sampling Normal Distribution from the summation data is calculated as well as each attribute. And then threshold levels are found by referring the probability on the baseline of Table.3.

As a result, quantitative layer divisions are carried out. 130 athletes in Layer4 are unset for the analysis and 209 athletes above the line are settled for that. Furthermore, above the line, we can find the difference like Layer1 (22 athletes) as obvious "Head", Layer2 (59 athletes) as "nearly Head" and Layer3 (128 athletes) as "nearly Tail".

## 2.4 Clustering within Each Layer

After Creating Layers, we could divide all athletes into four layers based on quantitative social involvement. In the next process, qualitative classification is required in each layer above Layer3. The qualitative aspect of this data is deference of four social involvements. Therefore input attributes are the normalized values of (A)Acknowledgement, (B)Expectation, (C)Spectator, and (D)Satisfaction.

The algorithm of clustering applied for this data was SOM. According to the optimized index of Viscovery SOMine 3.0J, one of SOM software, we achieved the

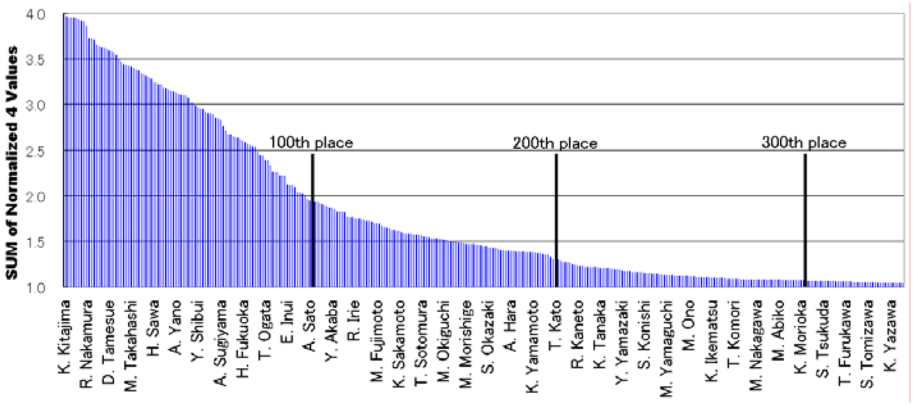


**Fig. 1.** Descending order of the summation of normalized four values of all athletes. The name of athletes at every nine interval in the horizontal axis.

**Table 3.** Threshold Levels of Layer Division and the Number of Athletes in each Layer

| Layer | Threshold Levels of Cumulative Probability | Number of Athletes | |
|---|---|---|---|
| 1 | Top 2.5% | 22 | Objecets of the Analysis<br>209 |
| 2 | Top 25%, Less than 2.5% | 59 | |
| 3 | Top 75%, Less than 25% | 128 | |
| 4 | Less than Top 75% | 130 | |
| SUM | | 339 | |

clustering. The result is that 12 clusters made of 22 athletes in Layer1, 23 clusters made of 59 athletes in Layer2, and 38 clusters made of 128 athletes in Layer3.

Following four figures (Fig.2, Fig.3, Fig.4 and Fig.5) are the correspondence between cluster and four input attributes in Layer1. In each figure, the warm color shows relatively high involvement; in contraries the cool color shows relatively low involvement in the layer. Omitting inserting figures, there are similar SOMs in Layer2 and Layer3 originally.

## 2.5 Layer Integration

Although Creating Layers was done by necessity owing the Long Tail behavior of the data, the goal of this analysis is to capture the general architecture of athlete evaluation. It is indispensable to integrate all layers. Concrete procedures to achieve that integration are the following.

First, each normalized involvement values ((A), (B), (C), and (D)) are averaged per cluster. Second, constituent ratios of four attributes in the sum total of normalized them as 100% are calculated within cluster. Third, the Similarity Index of clusters is

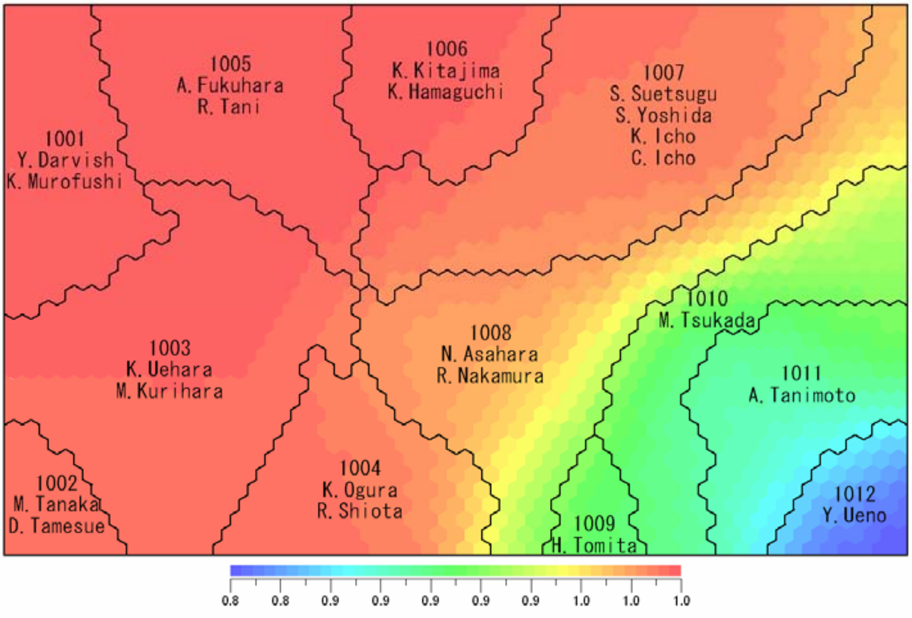


**Fig. 2.** Component Map of (A)Acknowledgement Attribute

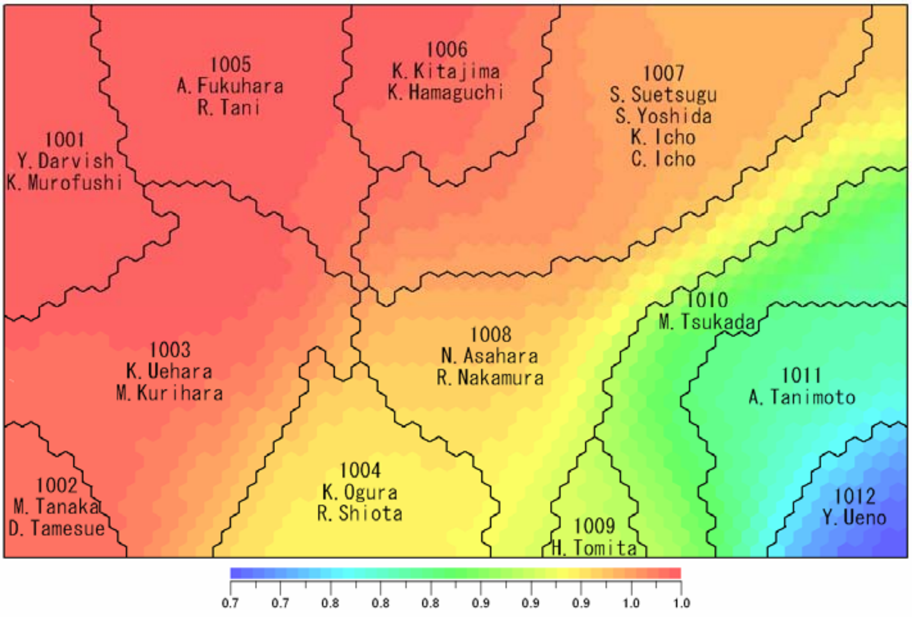


**Fig. 3.** Component Map of (B)Expectation Attribute

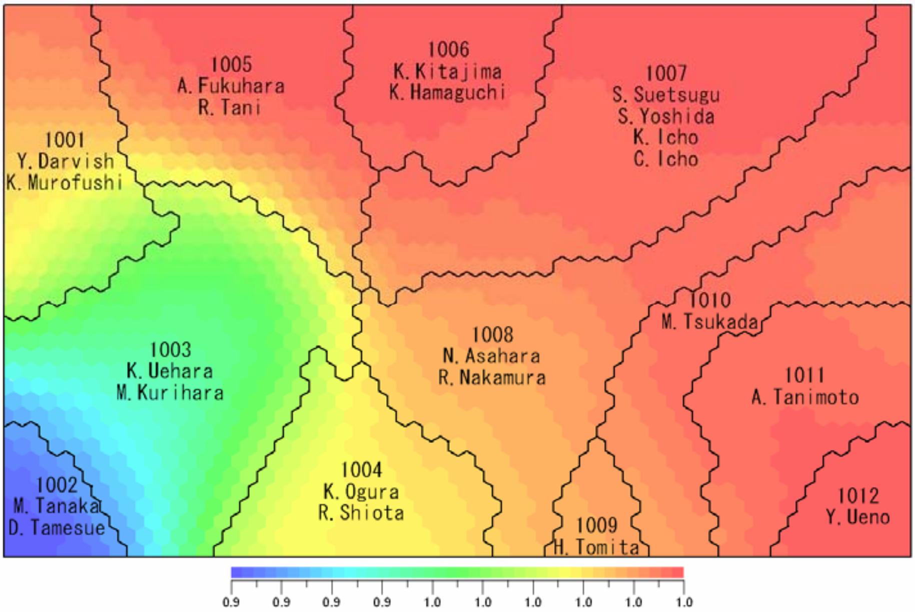


**Fig. 4.** Component Map of (C)Spectator Attribute

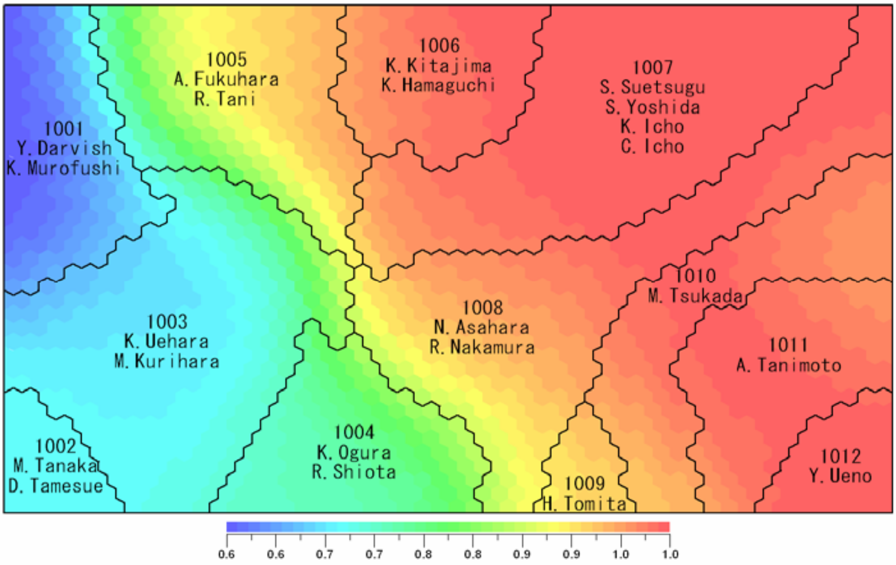


**Fig. 5.** Component Map of (D)Satisfaction Attribute

required between other layer's clusters by Formula(1)[1]. Forth, according to the Similarity Index, clusters indicating the highest value nearby layers are integrated.

After above procedures, Tree Diagrams appear. Though the correct number of Tree Diagrams is twelve, referring to the Similarity Index within Layer1, finally six evaluation patterns are extracted.

$$\text{Similarity Index} = \left( 1.00 - \sum_{i=1}^{k} \left| f_{1i} - f_{2i} \right| \right) * 100 \qquad (1)$$

## 3 Result

Final result is Table.4 and Table.5. Features of these tables are the following. Display sequence of athletes in a Layer is going along with descending order of summation of normalized values of four attributes. In the column of Sex, M means "male" and F means "female". Digits in the column of Rank indicate the best rank of the athlete in the 2008 Beijing Olympics[2]. If the value is "-", it means "did not play" or lost during the elimination. In addition, the inner figures inserting upper side of these figures are simplified images of the composition of four attributes.

To begin with overviewing the table, in the point of involvement disproportion between Before Olympics ((A) and (B)) and After Olympics ((C) and (D)), the evaluation pattern, which has equivalent involvements, is just only Pattern4. Other five patterns are lopsided one side. Table.4 shows the result of the types of (A) and (B) are high but (C) and (D) are low in comparison. Table.5 excepting Pattern4 is the result of the opposite types or approximately-same ratio type between four attributes.

In the next place, in the point of head-count, the minimum size pattern is Pattern.4. It has no more than 8 athletes in Layer1. In the other patterns, the more going to left side of Table.4 and right side of Table.5, the more number of the athletes belong. It does not make much difference between biased Before Olympics (Pattern1, Pattern2 and Pattern3) and After Olympics (Pattern5 and Pattern6) in total. However, the composition of belonging layers is definitely different. Especially Pattern1 has abundant Layer2 (33 athletes), on the other hand, Pattern6 has abundant Layer3 (68 athletes).

The following findings were obtained from considerations of the results. The athletes with high performance, their expectation are high (Pattern1). A sport without highly expected players, might face industrial reduction in future (Pattern2). There are peculiar evaluation aspects toward female athletes (Pattern3). The scarcity of athletes who are both high expectation and good result (Pattern4). The difficulty of continuing social involvement for athletes (Pattern5). The possibility of expanding social involvement rapidly, and the problems of concentrating social concern towards particular players (Pattern6), and so on.

[1] $f_{1i}$ is a constituent ratio of attribute $i$ of a cluster and $f_{2i}$ is a constituent ratio of same attribute $i$ of another cluster. $k$ indicates the number of attribute (4 in this case).

[2] Ranks are in the point of time of Ending Beijing Olympics. It is thought the case of changing ranks. One of the examples is a disqualification of doping of the medalists.

**Table 4.** Three Evaluation Patterns of Japanese Representative Athletes whose social involvement are biased Before the 2008 Beijing Olympics

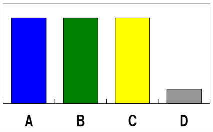
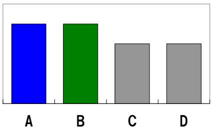
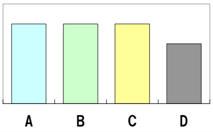
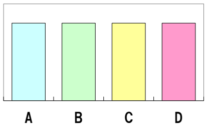
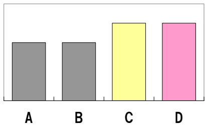
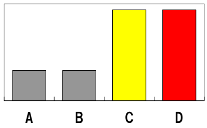
| Layer | Pattern1 | | | | Pattern2 | | | | Pattern3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| |  | | | |  | | | |  | | | |
| | Athelete | Sex | Event | Rank | Athelete | Sex | Event | Rank | Athelete | Sex | Event | Rank |
| 1 | Y. Darvish | M | Baseball | 4 | K. Uehara | M | Baseball | 4 | A. Fukuhara | F | TableTennis | 4 |
| | K. Murofushi | M | Field | 5 | M. Tanaka | M | Baseball | 4 | R. Tani | F | Judo | 3 |
| | | | | | M. Kurihara | F | Volleyball | 5 | K. Ogura | F | Badminton | 5 |
| | | | | | D. Tamesue | M | Track | 14 | R. Shiota | F | Badminton | 5 |
| 2 | R. Tosa | F | Marathon | – | A. Inaba | M | Baseball | 4 | H. Miyake | F | Weightlifting | 6 |
| | K. Kawakami | M | Baseball | 4 | N. Aoki | M | Baseball | 4 | Y. Nakanisih | F | Swimming | 18 |
| | S. Abe | M | Baseball | 4 | Y. Naruse | M | Baseball | 4 | | | | |
| | Y. Takeshita | F | Volleyball | 5 | S. Sugiyama | F | Volleyball | 5 | | | | |
| | M. Takahashi | F | Volleyball | 5 | K. Ikeda | F | Field | 20 | | | | |
| | K. Fujikawa | M | Baseball | 4 | E. Araki | F | Volleyball | 5 | | | | |
| | A. Shibata | F | Swimming | 27 | Y. Sano | F | Volleyball | 5 | | | | |
| | K. Suzuki | M | Judo | – | H. Izumi | M | Judo | – | | | | |
| | T. Wada | M | Baseball | 4 | | | | | | | | |
| | S. Miyamoto | M | Baseball | 4 | | | | | | | | |
| | K. Fukushi | F | Track | 11 | | | | | | | | |
| | H. Iwase | M | Baseball | 4 | | | | | | | | |
| | T. Sugiuchi | M | Baseball | 4 | | | | | | | | |
| | M. Kawasaki | M | Baseball | 4 | | | | | | | | |
| | S. Kimura | F | Volleyball | 5 | | | | | | | | |
| | A. Yano | M | Baseball | 4 | | | | | | | | |
| | T. Arai | M | Baseball | 4 | | | | | | | | |
| | T. Satozaki | M | Baseball | 4 | | | | | | | | |
| | T. Nishioka | M | Baseball | 4 | | | | | | | | |
| | H. Wakui | M | Baseball | 4 | | | | | | | | |
| | G.G.Sato | M | Baseball | 4 | | | | | | | | |
| | M. Araki | M | Baseball | 4 | | | | | | | | |
| | K. Omura | F | Volleyball | 5 | | | | | | | | |
| | Y. Shibui | F | Track | 17 | | | | | | | | |
| | S. Murata | M | Baseball | 4 | | | | | | | | |
| | A. Tajimi | F | Volleyball | 5 | | | | | | | | |
| | A. Sugiyama | F | Tennis | 32 | | | | | | | | |
| | H. Nakajima | M | Baseball | 4 | | | | | | | | |
| | M. Morino | M | Baseball | 4 | | | | | | | | |
| | M. Noguchi | F | Marathon | – | | | | | | | | |
| | K. Nishikiori | M | Tennis | 64 | | | | | | | | |
| | S. Hagino | M | Volleyball | 11 | | | | | | | | |
| | Y. Okamoto | F | TaeKwonDo | – | | | | | | | | |
| 3 | M. Saeki | F | BeachVolley | 19 | T. Ogata | M | Marathon | 13 | M. Taneda | F | Swimming | 8 |
| | T. Yamamoto | M | Volleyball | 11 | A. Sato | M | Marathon | 76 | K. Terauchi | M | Diving | 11 |
| | Y. Koshigawa | M | Volleyball | 11 | T. Morita | M | Swimming | 10 | Y. Kobayashi | F | Track | 16 |
| | T. Lee | M | Football | 15 | H. Ito | F | Swimming | 8 | A. Tanno | F | Track | 15 |
| | T. Morimoto | M | Football | 15 | K. Honda | M | Football | 15 | Y. Suenaga | M | Swimming | 13 |
| | Y. Sakurai | F | Volleyball | 5 | A. Uchida | M | Football | 15 | M. Yoshida | M | Football | 15 |
| | Y. Ishijima | M | Volleyball | 11 | M. Kano | F | Volleyball | 5 | T. Matsumiya | M | Track | 31 |
| | D. Usami | M | Volleyball | 11 | M. Yasuda | M | Football | 15 | S. Aoki | F | Track | 15 |
| | H. Hokatsu | M | Equitation | 10 | H. Mizumoto | M | Football | 15 | | | | |
| | N. Saito | M | Volleyball | 11 | Y. Kawai | F | Volleyball | 5 | | | | |
| | K. Yamamura | M | Volleyball | 11 | T. Honda | M | Football | 15 | | | | |
| | K. Asahi | M | BeachVolley | 9 | S. Nishikawa | M | Football | 15 | | | | |
| | C. Fukushima | F | Track | 47 | K. Shimizu | M | Volleyball | 11 | | | | |
| | A. Morita | F | Tennis | 32 | C. Kusuhara | F | BeachVolley | 19 | | | | |
| | | | | | S. Kagawa | M | Football | 15 | | | | |
| | | | | | Y. Nagatomo | M | Football | 15 | | | | |
| | | | | | K. Tsumagari | M | Volleyball | 11 | | | | |
| | | | | | K. Tomonaga | M | Volleyball | 11 | | | | |
| | | | | | D. Sawano | M | Field | 16 | | | | |
| | | | | | S. Osaki | M | Marathon | – | | | | |
| | | | | | K. Nasusako | M | Track | 14 | | | | |
| | | | | | H. Hosogai | M | Football | 15 | | | | |
| | | | | | M. Morishige | M | Football | 15 | | | | |
| | | | | | Y. Kajiyama | M | Football | 15 | | | | |
| | | | | | H. Hirota | F | Trampoline | 12 | | | | |
| | | | | | Y. Matsumoto | M | Volleyball | 11 | | | | |
| | | | | | S. Okazaki | M | Football | 15 | | | | |
| | | | | | Y. Toyoda | M | Football | 15 | | | | |
| | | | | | H. Taniguchi | M | Football | 15 | | | | |
| | | | | | M. Hayakari | F | Track | 35 | | | | |
| | | | | | Y. Kanemaru | M | Track | 43 | | | | |
| | | | | | K. Yamamoto | M | Football | 15 | | | | |
| | | | | | T. Nakatsuka | M | Cycling | 6 | | | | |
| | | | | | T. Fukuzawa | M | Volleyball | 11 | | | | |
| | | | | | T. Fushimi | M | Cycling | 21 | | | | |

**Table 5.** Three Evaluation Patterns of Japanese Representative Athletes whose social involvement are balanced or biased After the 2008 Beijing Olympics

| Layer | Pattern4 | | | | Pattern5 | | | | Pattern6 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| |  | | | |  | | | |  | | | |
| | Athelete | Sex | Event | Rank | Athelete | Sex | Event | Rank | Athelete | Sex | Event | Rank |
| 1 | K. Kitajima | M | Swimming | 1 | A. Tanimoto | F | Judo | 1 | Y. Ueno | F | Softball | 1 |
| | S. Suetsugu | M | Track | 3 | H. Tomita | M | Gymnastics | 2 | | | | |
| | K. Hamaguchi | F | Wrestling | 3 | M. Tsukada | F | Judo | 2 | | | | |
| | S. Yoshida | F | Wrestling | 1 | | | | | | | | |
| | K. Icho | F | Wrestling | 1 | | | | | | | | |
| | C. Icho | F | Wrestling | 2 | | | | | | | | |
| | N. Asahara | M | Track | 3 | | | | | | | | |
| | R. Nakamura | F | Swimming | 3 | | | | | | | | |
| 2 | | | | | Y. Nakamura | F | Marathon | 13 | M. ueno | F | Judo | 1 |
| | | | | | | | | | H. Sawa | F | Football | 4 |
| | | | | | | | | | N. Tsukahara | M | Track | 3 |
| | | | | | | | | | M. Uchishiba | M | Judo | 1 |
| | | | | | | | | | S. Hirano | F | TableTennis | 4 |
| | | | | | | | | | S. Ishii | M | Judo | 1 |
| | | | | | | | | | T. Matsuda | M | Swimming | 3 |
| | | | | | | | | | K. Uchimura | M | Gymnastics | 2 |
| | | | | | | | | | M. Nakamura | F | Judo | 3 |
| | | | | | | | | | S. Harada | F | Synchronized | 3 |
| | | | | | | | | | E. Suzuki | F | Synchronized | 3 |
| | | | | | | | | | H. Fukuoka | F | TableTennis | 4 |
| | | | | | | | | | S. Takahira | M | Track | 3 |
| | | | | | | | | | S. Suetsuna | F | Badminton | 4 |
| | | | | | | | | | M. Maeda | F | Badminton | 4 |
| 3 | | | | | T. Kashima | M | Gymnastics | 2 | Y. Ota | M | Fencing | 2 |
| | | | | | Y. Akaba | F | Track | 20 | E. Arakawa | F | Football | 4 |
| | | | | | R. Irie | M | Swimming | 5 | R. Nishiyama | F | Softball | 1 |
| | | | | | H. Hiraoka | M | Judo | − | E. Yamada | F | Softball | 1 |
| | | | | | Y. Kanemaru | M | Judo | 7 | E. Inui | F | Softball | 1 |
| | | | | | R. Utsugi | F | Football | 4 | H. Sakai | F | Softball | 1 |
| | | | | | T. Ono | M | Judo | − | Y. Mine | F | Softball | 1 |
| | | | | | Y. Okumura | M | Swimming | 7 | A. Sato | F | Judo | 7 |
| | | | | | Y. Ueyama | M | Trampoline | 9 | M. Mishina | F | Softball | 1 |
| | | | | | M. Nakagawa | F | Diving | 11 | S. Mabushi | F | Softball | 1 |
| | | | | | M. Mita | F | Swimming | 7 | R. Sato | F | Softball | 1 |
| | | | | | R. Kaneto | F | Swimming | 7 | N. Emoto | F | Softball | 1 |
| | | | | | S. Kubokura | F | Track | 15 | M. Hirose | F | Softball | 1 |
| | | | | | | | | | A. Karino | F | Softball | 1 |
| | | | | | | | | | M. Someya | F | Softball | 1 |
| | | | | | | | | | S. Ito | F | Softball | 1 |
| | | | | | | | | | S. Nakazawa | F | Judo | − |
| | | | | | | | | | K. Tsurumi | F | Gymnastics | 5 |
| | | | | | | | | | M. Fukumoto | F | Football | 4 |
| | | | | | | | | | M. Fujimoto | F | Softball | 1 |
| | | | | | | | | | S.Ohno | F | Football | 4 |
| | | | | | | | | | Y. Kan | M | TableTennis | 5 |
| | | | | | | | | | J. Mizutani | M | TableTennis | 5 |
| | | | | | | | | | K. Sakamoto | M | Gymnastics | 2 |
| | | | | | | | | | T. Sotomura | M | Trampoline | 4 |
| | | | | | | | | | H. Ikeda | F | Football | 4 |
| | | | | | | | | | Y. Nagasato | F | Football | 4 |
| | | | | | | | | | K Ando | F | Football | 4 |
| | | | | | | | | | J. Miyashita | M | Swimming | 3 |
| | | | | | | | | | A. Miyama | F | Football | 4 |
| | | | | | | | | | M. Okiguchi | M | Gymnastics | 2 |
| | | | | | | | | | K. Maruyama | F | Football | 4 |
| | | | | | | | | | Y. Kinga | F | Football | 4 |
| | | | | | | | | | S. Kishikawa | M | TableTennis | 5 |
| | | | | | | | | | A. Kitagawa | F | Swimming | 6 |
| | | | | | | | | | T. Matsunaga | M | Wrestling | 2 |
| | | | | | | | | | T. Nakase | M | Gymnastics | 2 |
| | | | | | | | | | N. Kawashima | F | Synchronized | 5 |
| | | | | | | | | | A. Iwashimizu | F | Football | 4 |
| | | | | | | | | | A. Aoki | F | Synchronized | 5 |
| | | | | | | | | | H. Kobayashi | F | Synchronized | 5 |
| | | | | | | | | | A. Matsumura | F | Synchronized | 5 |
| | | | | | | | | | M. Yanagita | F | Football | 4 |
| | | | | | | | | | E. Hirose | F | Badminton | − |
| | | | | | | | | | A. Hara | F | Football | 4 |
| | | | | | | | | | M. Tachibana | F | Synchronized | 5 |
| | | | | | | | | | K. Yumoto | M | Wrestling | 3 |
| | | | | | | | | | Y. Ishiguro | F | Synchronized | 5 |
| | | | | | | | | | K. Yano | F | Football | 4 |
| | | | | | | | | | E. Komura | F | Synchronized | 5 |
| | | | | | | | | | A. Kaihori | F | Football | 4 |
| | | | | | | | | | M. Sakaguchi | F | Football | 4 |
| | | | | | | | | | T. Kato | F | Football | 4 |
| | | | | | | | | | K. Oshima | F | Gymnastics | 5 |
| | | | | | | | | | T. Fujii | M | Swimming | 3 |
| | | | | | | | | | M. Uemura | F | Gymnastics | 5 |
| | | | | | | | | | N. Misawa | F | Rhythmic | 10 |
| | | | | | | | | | H. Sato | M | Swimming | 3 |

## 4 Conclusion and Future Work

Three implications are extracted from above. First is scarcity of proportion between "Expectation" and "Satisfaction". It means the severity of Olympic games for everyone including proven athletes. Second, the more brilliant career they have, the more disappointment risks they take. And finally, Olympics are an extraordinary chance for unknown athletes to upgrade their social values.

For separating out six evaluation patterns, it goes without saying that SOM performed a critical role. We could treat uncertain data of public opinion survey because of the flexibility of SOM represents undefined solution [7]. Furthermore, the evaluation patterns could be deciphered concretely because of the clever point of SOM, or visualization like Fig.2 to Fig.5. Therefore SOM is the indispensable method for this model.

We can say that this research established the model of capturing the changes of public opinion toward athletes in time series at least in the 2008 Beijing Olympic games. Additionally these evaluation patterns assumedly suit for various kind of sports events. In general, this method would apply to some other evaluation of media communication between performers and audience, such as actresses/actors on movies or drama programs, comedians on comedy-shows, or singers on music-programs.

However, this accomplishment is limited to the 2008 Beijing Olympics. So it is necessary to accumulate case examples for generalization. These cases are not only Summer Olympics but also Winter ones, not only Olympics but also other international athletic meeting. Both possibility and limitation of this model should be found, and adding improvements on them is the future work.

## References

1. Cardozo, R.N.: An Experimental Study of Consumer Effect, Expectation, and Satisfaction. Journal of Marketing Research 2, 244–249 (1965)
2. Oliver, R.L.: A Cognitive Model of the Antecedents and Consequences of Satisfaction Decisions. Journal of Marketing Research 17, 449–460 (1980)
3. Anderson, C.: The Long Tail. Wired Magazine (2004), `http://www.wired.com/wired/archive/12.10/tail.html`
4. Onoda, T.: Soft-Structuring Model:Supporting Category Management of the Long Tail. Ph.D Thesis, Graduate School of Media and Governance, Keio University (2007)
5. Kohonen, T.: Self-Organized Formation of Topology Correct Feature Maps. Biol. Cybern. 43, 59–69 (1982)
6. Japanese Olympic Committee: Japanese delegation recognition list (2008), `http://www.joc.or.jp/beijing/athlete/index.html`
7. Tokutaka, H., Fujimura, K.: Self-Organizing Maps (SOM) and their Applications. Journal of Japan Society for Fuzzy Theory and Systems 13(4), 3–13 (2001)

# Temporal Signal Processing by Feedback SOM: An Application to On-line Character Recognition Task

Hiroshi Wakuya and Akira Terada

Faculty of Science and Engineering, Saga University,
Honjo-machi, Saga 840-8502, Japan
wakuya@ee.saga-u.ac.jp, terada@dna.ec.saga-u.ac.jp

**Abstract.** An Elman-type feedback SOM (EFSOM) is a revised version of the standard SOM for dealing with time-variant information. In order to estimate its performance, an on-line character recognition task is tried with a large-scale training data set in this paper. At the same time, an idea for increasing the number of firing neurons in the competitive layer is examined to improve its trainability by means of enriching the state information. After the EFSOM is trained successfully, some validation tests for spatial displacement and temporal elasticity are also investigated. As a result of computer simulations, it is found that the EFSOM shows good performance and has good temporal signal processing ability.

**Keywords:** Feedback SOM, Temporal signal processing, On-line character recognition, Spatial displacement, Temporal elasticity.

## 1 Introduction

A self-organizing map (SOM) [1] is a neural network model for the visual information processing system in the biological brain. In spite of such origin, it is used widely in various kinds of fields as a good tool for signal converter these days. Its major reason is a topology-preserved projection from the input layer to the competitive layer developed through training. But one of its drawbacks is that there are no abilities to deal with time-variant information. Then, in order to overcome the defect, several nice ideas are introduced by a lot of researchers [2,3,4,5]. Among them, an Elman-type feedback SOM (EFSOM) is adopted in this study, and its temporal signal processing ability is investigated with an on-line character recognition task.

In the following part of this paper, the EFSOM is explained in Section 2. Secondly, some computer simulations are performed in Section 3. Thirdly, discussion from the viewpoints of the further considerations is made in Section 4. Finally, conclusions from this study are summarized in Section 5.

## 2 Elman-Type Feedback SOM

An Elman-type feedback SOM (EFSOM) is a revised version for temporal signal processing. As shown in Fig.1, it is a standard SOM with a feedback pathway

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 865–873, 2009.

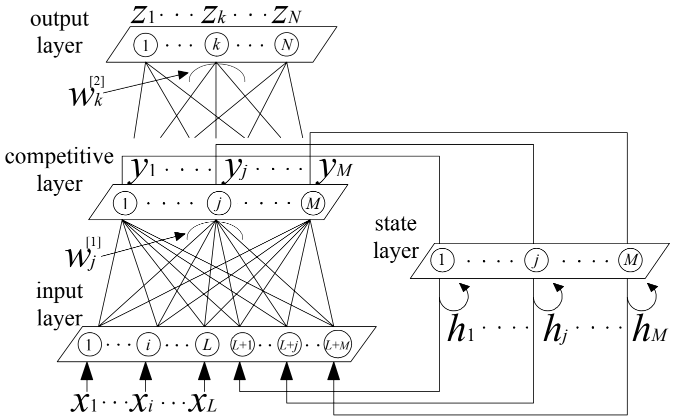

output layer
$Z_1 \cdots Z_k \cdots Z_N$
1 k N
$w_k^{[2]}$
competitive layer
$y_1 \cdots y_j \cdots y_M$
1 j M
$w_j^{[1]}$
state layer
1 j M
$h_1 \cdots h_j \cdots h_M$
input layer
1 i L L+1 L+j L+M
$x_1 \cdots x_i \cdots x_L$

**Fig. 1.** Elman-type feedback SOM

around the competitive layer, and also has an extra output layer to identify several candidates suitable for the applied pattern. Because its structure is similar to one of the famous recurrent neural networks, i.e., the Elman's model [6], it is called an Elman-type feedback SOM.

According to Fig.1, the EFSOM consists of four layers, i.e., an input layer $\boldsymbol{x}$, a hidden layer $\boldsymbol{y}$, a state layer $\boldsymbol{h}$, and an output layer $\boldsymbol{z}$. The numbers of neurons in each layer are represented by $(L+M)$, $M$, $M$, $N$, respectively. Also, weights from all neurons in the input layer to the $j$-th neuron in the competitive layer are denoted in the vector form as $\boldsymbol{w}_j^{[1]}$, and weights from all neurons in the competitive layer to the $k$-th neuron in the output layer are denoted as $\boldsymbol{w}_k^{[2]}$ similarly. Then, signal transmission equations are defined as follows:

$$d_j(t) = \sqrt{\sum_{i=1}^{L} \left\{ x_i(t) - w_{ji}^{[1]} \right\}^2 + \sum_{i=1}^{M} \left\{ \beta h_i(t) - w_{j,L+i}^{[1]} \right\}^2} \ , \tag{1}$$

$$j^* = \arg \min_{1 \leqq j \leqq M} d_j(t) \ , \tag{2}$$

$$y_j(t) = \begin{cases} 1.0 \ , & j = j^* \ , \\ 0 \ , & j \neq j^* \ , \end{cases} \tag{3}$$

$$h_j(t) = (1-\gamma) y_j(t) + \gamma h_j(t-1) \ , \tag{4}$$

$$z_k(t) = \sum_{j=1}^{M} w_{kj}^{[2]} y_j(t) \ , \tag{5}$$

where $j^*$ is an index number of the winner neuron, $\beta$ is a coefficient for referring the past history, and $\gamma$ is a coefficient for preserving the past history.

In the training mode, weight adaptation is divided into two major parts: One is for the weights $\boldsymbol{w}_j^{[1]}$, and the other is for the weights $\boldsymbol{w}_k^{[2]}$. The former part is the same as the standard SOM, i.e., just a repetition of the competitive and cooperative phases, while the latter part is the Hebbian learning between the winner neuron accompanied with neighboring neurons and the output neurons applied the teacher signal $d_k(t)$. Therefore, if the training mode is completed successfully, it is expected that the corresponding neuron for the applied pattern responds bigger than the other neurons in the output layer.

## 3 Computer Simulations

### 3.1 Summary of Preceding Studies

In general, previous studies on the EFSOM are carried out mainly with a small number of training samples. For example, as the first step, four city names in Japan represented by Braille[1] are recognized successfully [5]. Then, from the viewpoint aiming at the future development, a large-scale training data set is introduced as an upcoming problem.

According to the preceding study, it is clear that increasing the number of training samples makes training difficult. This is why the initial state of the EFSOM is the same as a start point for all training data, and it is a distinct nature for temporal signal processing. It means that weight updates are carried out for many times around the winner neuron corresponding to the initial state. This effect is naturally accumulated in proportion to the number of training samples. Then, depressing the training rate at the beginning of the training mode is important, and its effectiveness is confirmed through some computer simulations [7].

### 3.2 Overview of the Task, and Experimental Methods

Based on the prior knowledge mentioned above, a set consisting of 18 characters as shown in Fig.2 is adopted in this study. It is one of the Japanese native characters called “katakana”. Because each of them consists of semi-straight lines containing similar partial components, it must be a tough problem to discriminate them correctly. But it is true that the stroke order of characters might be a clue to identify, even though its appearance is similar.

By the way, the proposed EFSOM can deal with time-variant information appropriately with the help of the feedback pathway. It means ideally that, for example, the EFSOM has the ability to discriminate arbitrary two patterns whose present inputs are the same but their past histories are different. Therefore, an on-line character recognition task[2] is tried in this paper to evaluate the

[1] Braille is a character system designed for visually impaired people. Each character is represented by a combination of up to 6 points on 3x2 grid.

[2] Only instant positional information of the penpoint is considered. Others including any traces for written characters are not taken into account.

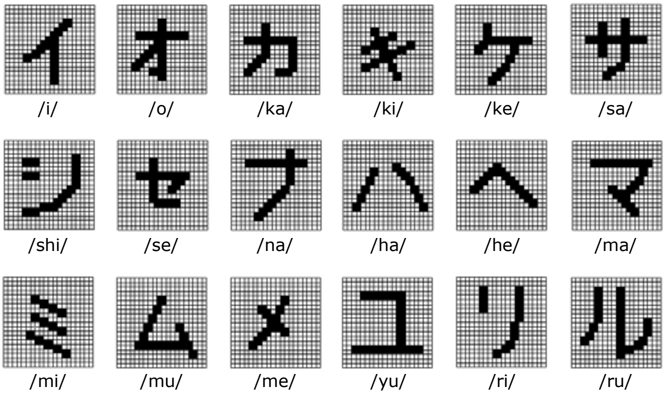


**Fig. 2.** A set of 18 characters "katakana" prepared for training

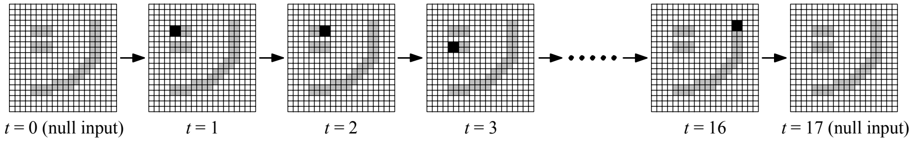


**Fig. 3.** An example of presenting a pattern `/shi/` to the EFSOM. A black area (2x2 dots) is applied at every instant $t$ except the start and end points.

EFSOM's performance. An example of presenting a pattern `/shi/` is explained in Fig.3. As can be seen in it, the size of each character is 20x20 dots wide. Among them, only a black area (2x2 dots) showing the penpoint information makes the corresponding input neurons active (1) at every instant $t$, while a white area makes them non-active (0). A gray area is a shape of the character for easy to identify.

Based on the task adopted in this study, the EFSOM is designed as,

$$L = 20 \times 20 = 400 \ , \qquad M = 30 \times 30 = 900 \ , \qquad N = 18 \ ,$$

$$\beta = 4.0 \ , \qquad \gamma = 0.5 \ .$$

From the viewpoint of the automaton-like system, each neuron in the competitive layer can represent different state information, so increasing the number of neurons $M$ must be a good strategy to improve the score. But another idea to utilize a combination of plural winner neurons, not a conventional single winner, is proposed. Then, (2) and (3) are replaced by the following equations:

$$j_1^* = \arg \min_{1 \leqq j \leqq M} d_j(t) \ , \tag{6}$$

$$j_2^* = \arg \min_{1 \leqq j \leqq M; j \neq j_1^*} d_j(t) \ , \tag{7}$$

$$j_3^* = \arg \min_{1 \leqq j \leqq M; j \neq j_1^*, j_2^*} d_j(t) \ , \tag{8}$$

$$y_j(t) = \begin{cases} 1.0 \ , & j = j_1^* \ , \\ 0.8 \ , & j = j_2^* \ , \\ 0.6 \ , & j = j_3^* \ , \\ 0 \ , & \text{others} \ . \end{cases} \tag{9}$$

Under the above-mentioned condition, each training procedure is repeated for 1000 epochs.

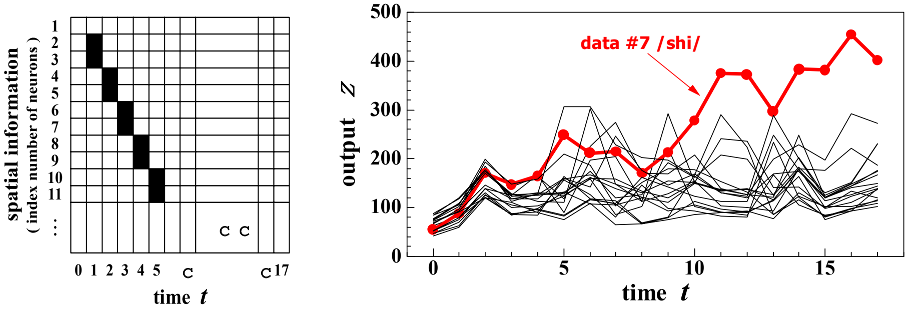
### 3.3 Experimental Results

**Validation Test for Training.** At first, all patterns shown in Fig.2 are applied to the trained EFSOM independently, and it is confirmed that they all are recognized successfully. One of the examples is shown in Fig.4. The left side version is a schematic image of a presenting pattern. In short, it is the penpoint spatial information changing with time[3]. On the other hand, the right side version is responses of the 18 neurons in the output layer. Among them, one thick line with the symbol (●) is behavior of the correct neuron, i.e., the corresponding neuron for the pattern applied to the EFSOM. The other 17 narrow lines are behavior of the remaining neurons. According to this figure, the correct neuron (`/shi/`) responds bigger and bigger with time $t$, and finally it becomes a dominant neuron in the output layer.

In order to make clear the mechanism of temporal signal processing, the trajectories of the three winner neurons $j_1^*$, $j_2^*$, and $j_3^*$ in the competitive layer are investigated. Figure 5 is the case for the pattern (`/shi/`), which is the same as Fig.4, and the three trajectories are marked by the different symbols (○, ●, ☆). In this figure, each number in the symbols shows the time $t$ when the winner neuron is emerged. At first glance, it is found that any winner neurons at the same instant often appear the close positions each other, and move together in the competitive layer. Taking into consideration of the topology-preserved projection, one of the major features the SOM possesses, this fact is easy to understand. Also, it is found that combining the winner neuron's order is effective and it makes the EFSOM enrich the state information, because some cases adopted the conventional single-winner neuron style are ended in failure.

**Validation Test for Spatial Displacement.** Figure 6 is the case for presenting the pattern (`/shi/`) displaced rightward only one dot. Roughly speaking, all lines in the training patterns are two dots wide, so half of them are still overlapped with the original training patterns as shown in Fig.6(a). The gray area in it is identical to the training pattern shown in Fig.4(a). According to Fig.6(b), the correct neuron (`/shi/`) responds relatively smaller than the normal case at the beginning, and later responds bigger gradually. In spite of such distortional behavior, it can be said generally that an overview of responses seems to be similar to that observed in Fig.4(b). An exceptional point is behavior of the neuron

[3] The role of Fig.4(a) is obvious when we compare it with upcoming Figs.6-7(a).



(a) a scheme of presentation (b) responses of the EFSOM

**Fig. 4.** Results of computer simulations [I] — Responses for the training pattern /shi/

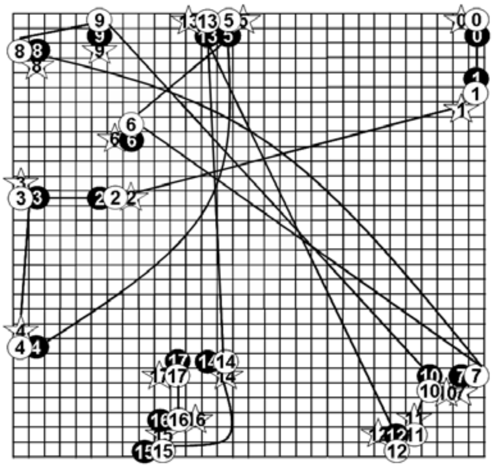
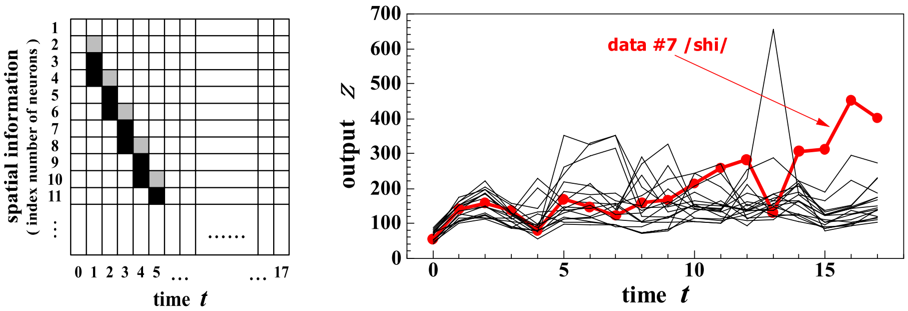


**Fig. 5.** Trajectories of the winner neurons $j_1^*$ (○), $j_2^*$ (●), and $j_3^*$ (☆) in the competitive layer. Each number in the symbols shows the time when the winner neuron is emerged.

(/ru/) at $t = 13$. It responds extremely bigger than the other neurons, and this fact reminds us that the other pattern sharing the similar penpoint positional information might attract to its corresponding state temporally.

By the way, all the patterns displaced four directions are also investigated as summarized in Table 1. Each check mark ($\surd$) represents that the EFSOM can recognize the corresponding one-dot-displaced pattern correctly. Also, the total evaluation is carried out based on the criterion that any of them is completed successfully. It is true that the total evaluation sometimes contains not so small deviation: Some of them (/shi/, /ru/) are succeeded in most directions, while others (/ki/, /ma/, /yu/, /ri/) are succeeded in only one direction.



(a) a scheme of presentation

(b) responses of the EFSOM

**Fig. 6.** Results of computer simulations [II] — Responses for the spatial displaced pattern `/shi/`. (An example of presenting only one dot rightward. )

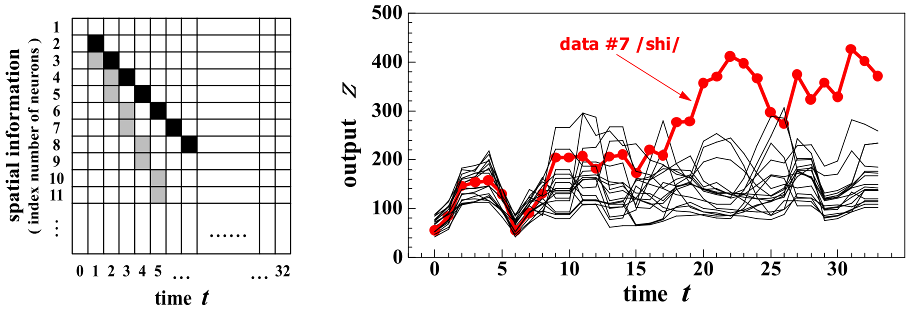
**Table 1.** Results of computer simulations for spatial displacement

| displacement | /i/ | /o/ | /ka/ | /ki/ | /ke/ | /sa/ | /shi/ | /se/ | /na/ |
|---|---|---|---|---|---|---|---|---|---|
| rightward | . | . | . | . | . | . | √ | . | . |
| leftward | . | . | . | . | . | . | √ | . | √ |
| upward | . | √ | . | √ | . | . | √ | . | √ |
| downward | . | √ | . | . | . | . | √ | . | . |
| total evaluation | . | √ | . | √ | . | . | √ | . | √ |

| displacement | /ha/ | /he/ | /ma/ | /mi/ | /mu/ | /me/ | /yu/ | /ri/ | /ru/ |
|---|---|---|---|---|---|---|---|---|---|
| rightward | . | . | . | . | . | √ | . | . | . |
| leftward | . | . | . | . | . | . | √ | √ | √ |
| upward | . | . | √ | . | . | . | . | . | √ |
| downward | . | . | . | . | . | √ | . | . | √ |
| total evaluation | . | . | √ | . | . | √ | √ | √ | √ |

Nevertheless, it is judged here that nine patterns out of 18 training ones are robust for spatial displacement.

**Validation Test for Temporal Elasticity.** As a next step, robustness for temporal elasticity is examined. The case for presenting the pattern (`/shi/`) at half the training speed is shown in Fig.7. All penpoint information (2x2) is divided into two pieces (2x1 + 2x1), and each of them is applied to the trained EFSOM one by one serially as shown in Fig.7(a). According to Fig.7(b), the correct neuron (`/shi/`) responds bigger and bigger with time $t$, and finally it becomes a dominant neuron in the output layer. This temporal development is quite similar to the case for the training pattern observed in Fig.4(b).



(a) an image of presentation (b) responses of the EFSOM

**Fig. 7.** Results of computer simulations [III] — Responses for the temporal elastic pattern `/shi/`. (An example of presenting at half the training speed. )

Strictly speaking, a validation test for temporal elasticity is defined as follows:

1. An applied pattern to the trained EFSOM at every instant is the same with the training one,
2. Only the difference between the training and the validation test is just a presenting period.

In this computer simulation, however, applied patterns are different from the training ones as mentioned above. Therefore, the validation test for temporal elasticity, which we have tried here, is not the one literally, but it is close to the validation test for spatial displacement. As if supporting this hypothesis, all nine patterns checked in the row “total evaluation” of Table 1, which are judged robust for *spatial displacement*, are also robust for *temporal elasticity*.

By the way, the other case for presenting the pattern (`/shi/`) at double the training speed is also investigated. Two successive penpoint information (2x2 + 2x2) are combined together (2x4), and it is applied to the trained EFSOM at a single instant. According to the result, not shown here for brevity, the same tendency as the case at half the speed is observed.

## 4 Discussion

In this paper, an idea to utilize a combination of plural winner neurons, not a conventional single winner, in the competitive layer is introduced. Its prime objective is to improve the trainability by means of increasing the number of firing neurons. But weight adaptation around the first winner neuron $j_1^*$ only is carried out during the training period, as the same as the conventional method. Paying attention to the advantages of multi-winner neuron style, weight adaptation around the second winner neuron $j_2^*$ and the third winner neuron $j_3^*$ must be worth thinking to improve the EFSOM's performance. But it is noticeable here that there seems to be some disadvantages of multi-winner neuron style, which are not expected in this paper, so careful considerations are required before its practical application.

## 5 Conclusions

In this paper, the Elman-type feedback SOM (EFSOM) is investigated to make clear its temporal signal processing ability. As a result of some computer simulations, it is found that the EFSOM shows good performance and has good temporal signal processing ability.

## Acknowledgment

This work was partially supported by a Grant-in-Aid for Scientific Research (C) No.20500208 from the Japan Society for the Promotion of Science.

## References

1. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biol. Cybern. 43, 59–69 (1982)
2. Chappell, G.T., Taylor, J.G.: The temporal Kohonen map. Neural Networks 6, 441–445 (1993)
3. Koskela, T., Varsta, M., Heikkonen, J., Kaski, K.: Temporal sequence processing using recurrent SOM. In: Proc. 2nd Int. Conf. on Knowledge-Based Intelligent Electronic Systems, vol. 1, B-1, pp. 278–290 (1998)
4. Horio, K., Yamakawa, T.: Feedback self-organizing map and its application to spatio-temporal pattern classification. Int. J. of Computational Intelligence and Applications 1, 1–18 (2001)
5. Wakuya, H., Harada, H., Shida, K.: An architecture of self-organizing map for temporal signal processing and its application to a Braille recognition task. IEICE Trans. Inf. & Syst (Japanese Edition) J87-D-II, 884–892 (2004) (in Japanese)[4]
6. Elman, J.L.: Finding structure in time. Cognitive Sci. 14, 179–211 (1990)
7. Wakuya, H., Noda, T.: A trial of on-line character recognition with a feedback SOM — Part-II Revision of training method for large data set —. In: Proc. 24th Fuzzy System Symposium, TF4-3, pp. 750–755 (2008) (in Japanese)

[4] Wakuya, H., Harada, H., Shida, K.: An architecture of self-organizing map for temporal signal processing and its application to a Braille recognition task. Syst. Comp. Jpn. 38, 62–71, John Wiley & Sons (2007)

# A Study on Clustering Method by Self-Organizing Map and Information Criteria

Satoru Kato[1], Tadashi Horiuchi[1], and Yoshio Itoh[2]

[1] Matsue College of Technology, 14-4 Nishi-ikuma, Matsue,
Shimane 690-8518, Japan
kato@matsue-ct.ac.jp

[2] Tottori University, 4-101 Koyama-cho minami,
Tottori 680-8550, Japan

**Abstract.** In this paper, we propose a clustering method by SOM and information criteria. In this method, initial cluster-candidates are derived by SOM, and then these candidates are merged appropriately based on information criterion such as BIC or AIC (Akaike Information Criterion). Through the clustering experiments for the artificial datasets and UCI Machine Learning Repository's datasets, we confirm that our proposed method can extract clusters more accurately and stably than the SOM-only method.

## 1 Introduction

Clustering by Self-Organizing Map (SOM[1]) can extract clusters of arbitrary distribution shapes based on the distance between the code-vectors (representative points of the input data)[2]. In recent, there are several improvemental methods which alter the basic SOM algorithm [3][4]. Hence, this is one of the "distance-based" clustering approaches. On the other hand, there are "distribution-based" clustering approaches that consider the distribution of input data when extracting clusters appropriately. For example, $x$-means method [5] adopts Bayesian Information Criterion (BIC) into $k$-means method.

Information criteria are also easily introduced into the clustering method by SOM. In this paper, we propose a clustering method by SOM and information criteria. In the proposing method, initial cluster-candidates are derived by SOM, and then these candidates are merged appropriately based on the information criterion such as BIC or AIC (Akaike Information Criterion).

Through the clustering experiments for the artificial datasets and UCI Machine Learning Repository's datasets, we confirm that our proposed method can extract clusters more accurately and stably than the SOM-only method. Furthermore, we show that AIC is suitable for the proposed method compared to BIC.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 874–881, 2009.

## 2 Clustering by SOM

### 2.1 Basic SOM Algorithm

SOM, proposed by Kohonen, is configured as shown in Fig.1. In the basic SOM learning algorithm [1], the code vectors are updated by using the following equations

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t)\, \Phi(p_i)\ (\mathbf{x} - \mathbf{w}_i(t)) \tag{1}$$

$$\Phi(p_i) = \exp\left(-\frac{{p_i}^2}{\sigma^2(t)}\right) \tag{2}$$

Here $\alpha(t)$ is the learning coefficient after $t$ learning steps. The coefficient starts from its initial value $\alpha_{\mathrm{ini}}$, and then decreases monotonically as $t$ increases, thus reaching its minimum at the pre-set maximum number of learning steps $T$. In addition, $\Phi(p_i)$ is a neighborhood function with the center at winner cell $c$, and $p_i$ is the distance from cell $i$ to the winner cell $c$ in the competitive layer. In Eq.2, $\sigma(t)$ is a time-varying parameter that defines the neighborhood size in the competitive layer. Like $\alpha(t)$ in Eq.1, this parameter decreases monotonically from $\sigma_{\mathrm{ini}}$ as learning proceeds.

As a result of learning, the similarity between learning data is expressed by the closeness on the grid in the competitive layer. In addition, the data density in the input data space is reflected in the distribution of code vectors after learning.

### 2.2 Cluster Extraction from SOM

In the maps built by SOM learning, the code vectors between adjacent cells in the grid of competitive layer are similar, and the data density in the input layer is reflected in distribution of code vectors after learning. Using these features, as pointed out by Terashima et al. [2], allows clustering by the detection of cluster boundaries as portions where the code vectors between adjacent cells are substantially different. The specific clustering procedure is presented below. In addition, one-dimensional SOM is used for simplicity of analysis; $m$ cells in the competitive layer are arranged in one-dimensional array.

1. Map building
   The input data are subjected to SOM learning to obtain a set of code vectors.
2. Map analysis
   (a) For every cell $i(i = 1, 2, ..., m-1)$, the code vector density $dW_i$ is found from following equations as the Euclidean distance between code vectors for cells $i$ and $i+1$:

$$dW_i = \parallel \mathbf{w}_i - \mathbf{w}_{i+1} \parallel \tag{3}$$

   (b) The code vector density $dW_i$ for every cell $i(i = 1, 2, ..., m-1)$ is normalized to its maximum and minimum in the range 0–1, thus obtaining the normalized density $dW'_i$:

$$dW'_i = \frac{dW_i - dW_{i\_\mathrm{min}}}{dW_{i\_\mathrm{max}} - dW_{i\_\mathrm{min}}} \tag{4}$$

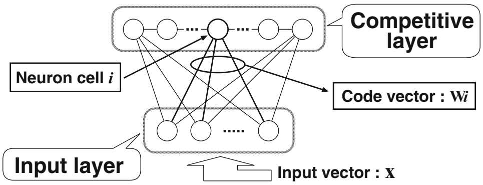


**Fig. 1.** Basic SOM structureione-dimensional SOMj

(c) The histogram of $dW_i'$ is derived. A cluster boundary is recognized between cell $i$ corresponding to the histogram peak and its neighbor cell $i+1$.

3. Labeling
   The competitive layer is divided according to the $dW_i'$ histogram, and every group of cells is labeled appropriately.

# 3 Proposed Method

## 3.1 Basic Idea

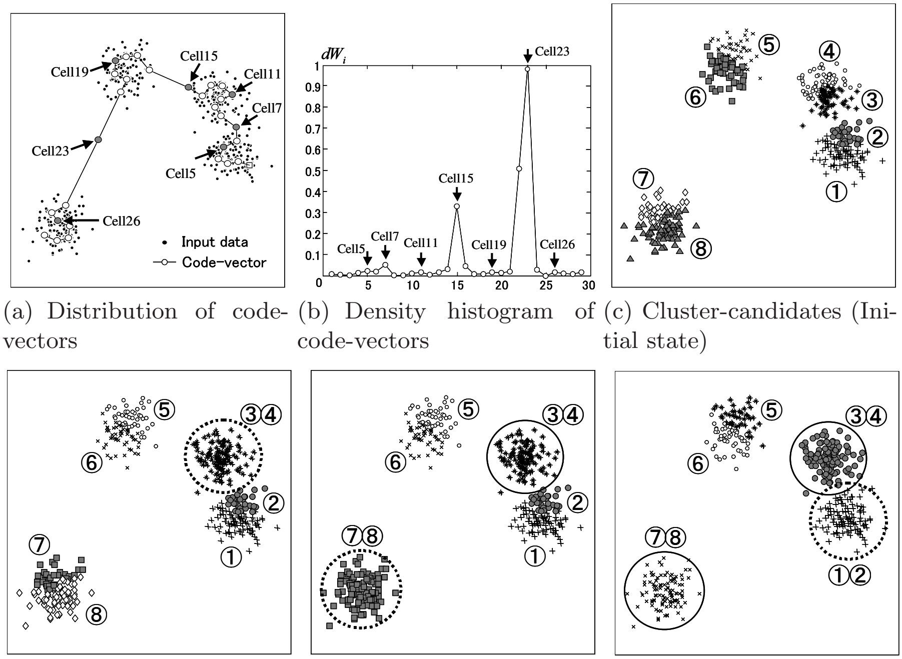
There are many upward-peaks in the density histogram of code-vector. Each of these may indicate a boundary of clusters obviously or not, so that many cluster-candidates can be extracted from the density histogram. Basic idea of the proposing method is appropriate mergence of these cluster-candidates by using of information criteria under following procedures.

A. Make code-vector density($dW_i'$) histogram after learning process of one-dimensional SOM.
B. Extract cluster-candidates from the density histogram and assign continuous number to each candidate. These numbers are ordered correspondingly to that of neuron cells in the competitive-layer of SOM.
C. Decide which cluster-candidates should be merged from arbitrary pair of candidates whose numbers are adjoining each other.

Fig.2 shows a practical sequence of the proposed method. Through the procedure A, we obtain Fig.2(a) and 2(b). And Fig.2(c) is obtained after the procedure B. Then, cluster-candidates are gradually merged by applying the procedure C over and over until the number of clusters is agree with original one. (See Fig.2(d)-2(f)).

## 3.2 BIC and AIC

When a distribution of data $\mathbf{x}$ is observed, a family of alternative model which generate the distribution can be considered. Information criterion is one of the



(a) Distribution of code-vectors (b) Density histogram of code-vectors (c) Cluster-candidates (Initial state)

(d) 1st. mergence (e) 2nd. mergence (f) 3rd. mergence

**Fig. 2.** Clustering process by using proposed method

useful guideline to determine which model is the most suitable. Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) are typical one, calculated by the following equation respectively.

$$\mathrm{BIC} = -2\log L(\hat{\theta};\mathbf{x}) + q\log n \tag{5}$$

$$\mathrm{AIC} = -2\log L(\hat{\theta};\mathbf{x}) + 2q \tag{6}$$

Where, $q$ is dimension of the parameter vector $\hat{\theta}$ and $n$ is the number of samples of empirical distribution. And $L(\cdot) = \prod f(\cdot)$ , where $f(\cdot)$ is $p$-dimensional Gaussian-distribution:

$$f(\hat{\theta};\mathbf{x}) =(2\pi)^{-\frac{p}{2}}|\mathbf{V}|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^{\mathrm{T}}\mathbf{V}^{-1}(\mathbf{x}-\mu)\right\} \tag{7}$$

In Eq.(5) and Eq.(6), first term is logarithmic likelihood when a model described by parameter $\hat{\theta}$ is applied to the empirical distribution of $\mathbf{x}$ and second term indicates complexity of the model which is called “penalty term”.

### 3.3 Cluster Mergence by Using Information Criterion

The procedure for selective cluster mergence (see Sec.3.1 procedure C) is divided into the following processes in practice. Here, it is assumed that the procedure A and B in Sec.3.1 have already finished.

C1. Merge a pair of adjoining cluster-candidates temporally, and calculate the two values what we call $\mathrm{IC}_{\mathrm{single}}$ and $\mathrm{IC}_{\mathrm{twin}}$ by using either Eq.(5) or Eq.(6). Where, $\mathrm{IC}_{\mathrm{single}}$ and $\mathrm{IC}_{\mathrm{twin}}$ means the value of either BIC or AIC when an applied distribution model for the unified clusters is single-distribution or twin-distribution respectively.

C2. Calculate $\Delta\mathrm{IC}$ which means a difference between $\mathrm{IC}_{\mathrm{single}}$ and $\mathrm{IC}_{\mathrm{twin}}$ by the following equation.

$$\Delta\mathrm{IC} = \mathrm{IC}_{\mathrm{single}} - \mathrm{IC}_{\mathrm{twin}} \tag{8}$$

C3. After calculation of $\Delta\mathrm{IC}$ for all pairs of adjoining cluster candidates, find the pair which has minimum $\Delta\mathrm{IC}$ and merge two cluster candidates included in the pair conclusively. Then, consecutive numbers for the cluster candidates including the new cluster are refreshed.

C4. Repeat the procedure C1 to C3 until the number of clusters reaches a specified value.

$\mathrm{IC}_{\mathrm{single}} < \mathrm{IC}_{\mathrm{twin}}$ when fitting the single-distribution to the unified clusters is more suitable than the case of twin-distribution. Therefore, $\Delta\mathrm{IC}$ can measure a degree of propriety to merge two adjoining clusters.

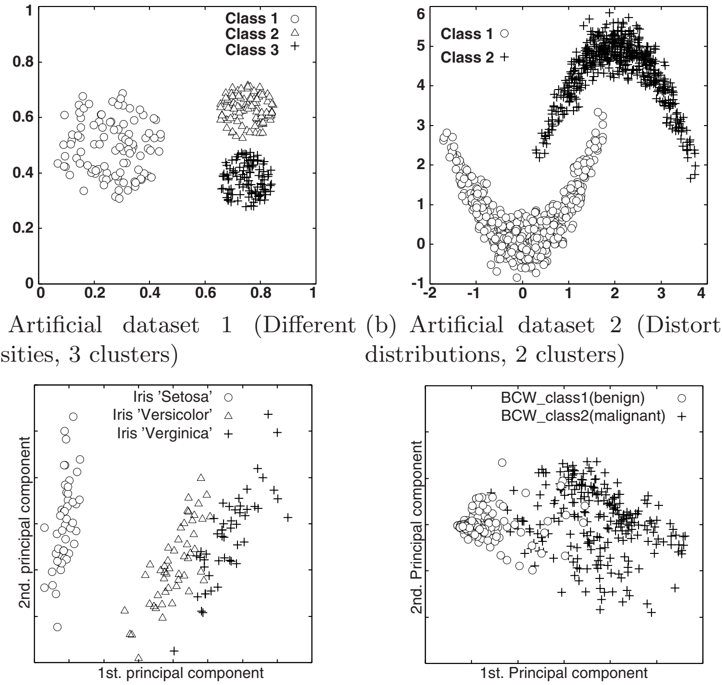
## 4 Clustering Experiments

### 4.1 Experimental Method

We use four kinds of data distribution as experimental dataset. Two datasets are generated artificially to consist of two or three clusters whose density or distribution shape is different. Each of another two datasets is UCI Iris and BCW dataset from UCI ML repository[6] as examples of practical data.

Performance evaluation is carried out by using a degree of the classification error. Classification error is calculated by comparing the indices of the original dataset with which is obtained from the clustering result.

When applying SOM learning algorithm in proposed method, we set the iteration of learning is 200 times of the number of the input data and the number of cells of the competitive layer is set to 15,20,25,30 or 35. We make 100 trials for each setting of SOM learning and apply cluster mergence procedure with either BIC or AIC for each SOM learning result so that 500 kinds of clustering result (100 trials × 5 patterns of the size of competitive layer) for each dataset.



(a) Artificial dataset 1 (Different densities, 3 clusters)

(b) Artificial dataset 2 (Distorted distributions, 2 clusters)

(c) UCI Iris data

(d) UCI BCW data

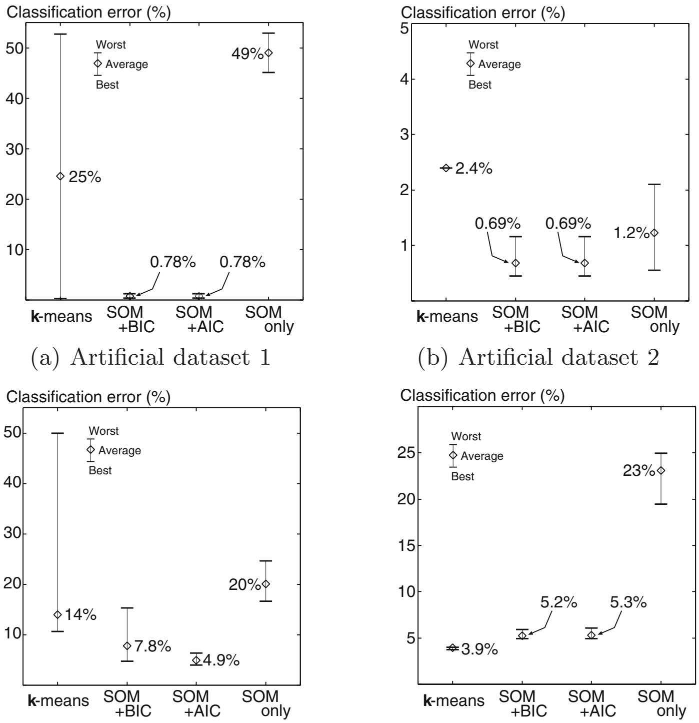
**Fig. 3.** Artificial and practical dataset for the clustering experiments

## 4.2 Experimental Result

Fig.4 shows the result of performance evaluation of clustering for each dataset. We calculate the average value of classification error through 100 trials for each the five pattern of SOM's competitive layer size. So that in the legend of each figure, "Worst","Average" and "Best" indicate maximum, average and minimum value of the average classification error respectively among the five patterns of SOM settings.

"SOM+BIC" and "SOM+AIC" correspond to proposed method and "SOM only" is conventional method which extracts clusters from histogram of code-vector density such as shown in Fig.2(b) with appropriate threshold setting.

In the case of artificial dataset1 and UCI BCW dataset, "SOM only" method shows very high classification error. These dataset are including clusters whose density is quite different each other and it is hard to estimate the boundary of clusters correctly only by using code-vector density histogram. On the other hand, proposed method can extract each cluster in the dataset more accurately than another methods except *k*-means method in the case of BCW dataset. BCW dataset contains comparatively high-dimensional data (each data has 10 attributes), therefore "distribution-based" approaches such as "SOM+BIC" and



(a) Artificial dataset 1

(b) Artificial dataset 2

(c) UCI Iris data

(d) UCI BCW data

**Fig. 4.** Comparison of clustering performance

"SOM+AIC" may not be able to estimate parameters such as $\mu$ and $\mathbf{V}$ in the Eq.(7) correctly of the distribution model.

Looking at the classification error of "SOM+BIC" and "SOM+AIC", both these methods show almost same clustering performance excepting the case of UCI Iris dataset. In Eq.(5), the penalty term includes variable of the number of samples $n$. And $\Delta$IC becomes small if the value of $n$ is large. Hence in the case of SOM+BIC method, one cluster candidate which has a large number of samples tends to take adjoining candidates one after another.

## 5 Conclusion

In this paper, we tried to combine SOM clustering methodology with appropriate cluster mergence approach by using information criteria such as BIC and AIC. Since it can pay attention to a naturalness for each data distribution as a cluster, proposed method can extract clusters more correctly than conventional methods especially when the dataset consists of clusters whose density is different each other.

From the results of clustering experiments using several kind of artificial and practical dataset, proposed method shows less classification error than other conventional method such as *k*-means and SOM-based simple clustering method. Furthermore, we confirmed that AIC is suitable for the proposed method compared to BIC.

It is necessary to use more kind of practical dataset for examination of effectiveness of the proposed method as a future work.

## References

1. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer, Heidelberg (2001)
2. Terashima, M., Shiratani, F., Yamamoto, K.: Unsupervised Cluster Segmentation Method Using Data Density Histogram on Self-Organizing Feature Map. IEICE Trans. J79-D-II(7), 1280–1290 (1996) (in Japanese)
3. Kato, S., Koike, K., Horiuchi, T., Itoh, Y.: A Study on Two-Stage Self-Organizing Map Suitable for Clustering Problems. In: Proceedings of the 2006 International Symposium on Intelligent Signal Processing and Communication Systems, pp. 677–680 (2006)
4. Matsushita, H., Nishio, Y.: Reunifying Self-Organizing Map and Disconnecting Self-Organizing Map. RISP Journal of Signal Processing 11(6), 445–452 (2007)
5. Pelleg, D., Moore, A.: *X*-means: Extending *K*-means with Efficient Estimation of the Number of Clusters. In: Proc. of the 17th International Conference on Machine Learning, pp. 727–734 (2000)
6. UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html

# Author Index